TECHNISCHE UNIVERSITÄT ILMENAU

Fakultät für Elektrotechnik und Informationstechnik

TENSOR-BASED SUBSPACE TRACKING

Master of Science in Communications and Signal Processing

Submitted by:	Olaa Khatib
Department:	Communications Research Laboratory
Major:	MSCSP
Responsible Professor:	Univ.-Prof. Dr.-Ing. Martin Haardt
Supervisor:	M. Sc. Yao Cheng
Date:	7. Februar 2014

URN: urn:nbn:de:gbv:ilm1-2014200011

ACKNOWLEDGEMENTS

I would like to thank Univ.-Prof. Dr.-Ing. Martin Haardt, M. Sc. Yao Cheng, Mr. Mark Martin, Dr. -Ing. Firas Shraif, Dr.sc.math. Abdull Nasser Shammsy and all my professors and teachers. Besides, I would like to thank my colleagues and friends. At the end, I would like to dedicate my work to my parents, brothers, sisters, children (Abdulallh and Abdulrahman), and my husband Aouss Gabash.

ABSTRACT

For different applications in the field of digital signal processing, subspaces estimation and tracking have been required, e.g., signal parameter estimation, data compressing, radar and imaging processing. One of the most fruitful techniques in estimating the signal subspaces is based on the singular value decomposition (SVD) concept. Recently, for multidimensional data, Higher-Order SVD (HOSVD) can be used to provide improved estimates of the subspace compared to the SVD concept. Moreover, the subspace estimates obtained by employing HOSVD can be used for parameter estimation in a harmonic retrieval problem where a multidimensional structure is inherent in the data. However, when the multidimensional data are time-variant, adaptive subspace tracking schemes based on tensor algebra are in demand. By employing the tensor-based subspace tracking algorithms, the signal parameters like DOA can be tracked as well. Moreover, if the number of observations is small or the sources are highly correlated, incorporating Forward Backward Averaging (FBA) can further improve the performance of tracking.

In this work, based on the tensor-based subspace tracking via Kronecker structured projections (TeTraKron) framework, we include FBA and propose the Extended FBA-PAST algorithm. We show that incorporating FBA leads to an improved accuracy of the subspace tracking and a lower computational complexity due to the fact that only real-valued processing is involved. Moreover, we evaluate the performances of the parameter estimation schemes in a variety of non-stationary scenarios where the subspace estimates are obtained by employing the subspace tracking algorithms. Furthermore, we extend the adaptive ESPRIT algorithm to a general case where the subarrays are not necessarily maximum overlapping. In addition, we develop an adaptive version of Unitary ESPRIT as well as 2-D Unitary ESPRIT. Compared to the direct combination of the PAST algorithm and Unitary ESPRIT or 2-D Unitary ESPRIT, the proposed adaptive schemes achieve the same performance with a lower mathematical complexity.

ZUSAMMENFASSUNG

Fr verschiedene Anwendungen auf dem Gebiet der digitalen Signalverarbeitung sind die Bestimmung der Unterrume sowie deren Tracking, zum Besispiel fr die Signalparameterschtzung, die Datenkomprimierung, Radar und die Bildverarbeitung, erforderlich. Eine der vielversprechendsten Techniken zur Schtzung der Signalunterrume basiert auf dem Konzept der Singulrwertzerlegung (Singular Value Decomposition, SVD). In letzter Zeit wurde fr mehrdimensionale Daten die SVD hherer Ordnung (Higher-Order SVD, HOSVD) verwendet, um verbesserte Schtzungen des Unterraums im Vergleich zum SVD-Konzept zu schaffen. Darber hinaus kann durch Verwendung der HOSVD die Schtzung des Unterraums fr die Parameterschtzung in einem harmonischen Wiedergewinnungsproblem mit mehrdimensionaler Struktur in den Daten, durchgefhrt werden. Sind jedoch die multidimensionalen Daten zeitvariant, werden adaptive Algorithmen, die auf der Tensoralgebra zum Tracking des Unterraums beruhen, bentigt. Durch den Einsatz dieser Algorithmen knnen auch die Signalparameter wie die Richtung (direction of arrival, DOA) bestimmt werden. Auerdem, wenn die Anzahl der Messungen gering ist oder die Quellen stark korreliert sind, kann dann durch die Anwendung der Vorwrts-Rckwrts-Durchschnittsbestimmung (Forward Backward Averaging, FBA) die Leistungsfhigkeit weiter verbessert werden. In dieser Arbeit bercksichtigen wir FBA und schlagen den erweiterten FBA-PAST-Algorithmus, der auf dem Tensor-Based Subspace Tracking via Kronecker structured projections (TeTraKron) basiert, vor. Wir zeigen, dass FBA zu einer verbesserten Genauigkeit des Unterraum-Tracking und einem niedrigeren Rechenaufwand durch reellwertige Rechenoperationen fhrt. Auerdem bewerten wir die Leistungsfhigkeit der Parameterschtzungsalgorithmen in vielen nicht-stationren Szenarien, in denen die Unterrume durch Verwendung des Unterraum-Tracking geschtzt werden. Darber hinaus erweitern wir den adaptiven ESPRIT-Algorithmus zu einem allgemeineren Fall, in dem die Unterarrays nicht notwendigerweise eine maximale berlappung haben. Weiterhin entwickeln wir eine adaptive Version fr Unitary ESPRIT sowie 2-D Unitary ESPRIT. Im Vergleich zu der direkten Kombination des PAST- Algorithmus mit Unitary ESPRIT oder 2-D Unitary ESPRIT, erreichen die vorgeschlagenen adaptiven Algortihmen die gleiche Leistung mit einer geringeren mathematischen Komplexitt.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Nowadays, tensors based subspace estimation and tracking are attracting more and more attention in numerous digital signal processing applications. In various signal processing tasks such as signal parameter estimation, data compressing, radar, imaging processing, the observation data are required to be stacked in multidimensional arrays. By increasing the storage media capacity, data are able to be stored more efficiently keeping the structure of data. In this regards, tensor algebra found its way as a promising tool for multidimensional arrays. For example, for a harmonic retrieval problem where the data are endowed with a multidimensional structure, suitable tools in tensor algebra can be used. One fruitful concept in linear algebra is the singular value decomposition (SVD). This concept has been extended to High-Order SVD (HOSVD) in [dLdMV00] depending on the Tucker model [Tuc66] for multidimensional data model. Tucker discussed the multilinear generalization of three-way data and derived a tensor decomposition in the SVD terminology. Afterward, De Lathauwer in [dLdMV00] used the term of HOSVD explaining the strong relation between the SVD and HOSVD concepts and giving the basics of tensor algebra. In order to calculate the HOSVD of N-order tensor, De Lathauwer identified terms of n-mode unfolding and a core tensor based on the SVD concept.

In tensor-based parameter estimation techniques, HOSVD is used to estimate the subspace as shown in [HRDG08]. More manipulation based on HOSVD has been performed according to [RBHW09a]. Roemer clarified that the signal subspaces can be calculated without computing the core tensor, which in turn reduces the mathematical complexity compare to the HOSVD.

Recently, [CH13] reduced the mathematical complexity further by using some properties in Kronecker products. [CH13] proposed a generic framework of tensor-based subspace tracking via Kronecker structured projections (TeTraKron). Here adaptive matrix-based subspace tracking algorithms, such as Projection Approximation Subspace Tracking (PAST) algorithm [Yan95] can be applied to calculate the subspaces of n-mode unfoldings. The PAST algorithm has been performed to track the signal subspaces if and only if the subspaces of signals are slowly changing. As aforementioned, [CH13] used this algorithm after modifying and extending it for a tensor-based

subspace tracking algorithm.

Tensor-based subspace tracking algorithm paved the way to track the signal parameters of multidimensional harmonic retrieval problems that needs an accurate knowledge of the signal subspaces.

Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [RPK86] is a high-resolution estimation technique that estimates the DOA with lower mathematical complexity compared to other algorithms such as Multiple Signal Classification (MUSIC) algorithm [Sch86]. ESPRIT has been applied to estimate the DOA based on the translation invariance structure of a sensor array.

By taking into account that the phase delay between two subarrays is unitary, [HN95] developed Unitary ESPRIT algorithm by incorporating Forward Backward Averaging (FBA). Besides, Unitary ESPRIT increased the accuracy of estimation. Moreover, it involved only real-valued computations reducing the mathematical complexity as well.

[HRDG08] extended both ESPRIT and Unitary ESPRIT algorithms for multidimensional data model to the Standard Tensor ESPRIT (STE) and Unitary Tensor ESPRIT (UTE) algorithms, respectively.

Returning back to the ESPRIT algorithm, [BRD03] modified it again for tracking the signal parameters based on the PAST algorithm. Here, Badeua reduced the mathematical complexity and kept the performance of the Adaptive ESPRIT as good as using the ESPRIT algorithm.

Our contribution in this thesis is to develop the aforementioned ideas in order to increase the tracking accuracy and to reduce the computational burden by incorporating the FBA processing.

The outline of thesis is organized as follows:

- the second chapter introduces an existing framework of tensor algebra and HOSVD. In addition, one application in using tensor-based subspace estimation in the digital signal processing field is given.
- The third chapter presents the performance of an adaptive version of tensor-based subspace tracking via Kronecker structured projections (TeTraKron) framework by incorporating FBA. And we show the performances of the parameter estimation schemes in a variety of non-stationary scenarios where the subspace estimates are obtained by employing the subspace tracking algorithms.

- The last chapter contains a proposal to extend the Adaptive ESPRIT algorithm to a general case where the subarrays are not necessarily maximum overlapping. In addition, we develop an adaptive version of Unitary ESPRIT as well as 2-D Unitary ESPRIT, and
- at the end of this thesis, we finish with conclusion and future work.

The notations throughout the work are as follows. To distinguish between scalars, vectors, matrices and tensors; italic letters $(a, b, ..., A, B)$, lower-case bold-faced letters $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{...})$, bold-faced capital letters $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{...})$, and bold-faced calligraphic letters $(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \boldsymbol{...})$, are used, respectively. The elements of matrix $\boldsymbol{A}$ and a tensor $\boldsymbol{\mathcal{B}}$ of third order are denoted as $a_{i,j}$ and $b_{i,j,k}$, respectively as well as $(\boldsymbol{A})_{i,j}$ and $(\boldsymbol{\mathcal{B}})_{i,j,k}$, respectively. The elements of a vector $\boldsymbol{a}$ are denoted similarly $(\boldsymbol{a})_i$. If necessary, we use MATLAB notation for denoting the matrix columns or rows, e.g., i-th row of a matrix $\boldsymbol{A}$ are written as $\boldsymbol{A}(i, :)$.

The superscripts ^{T}, ^{H}, *, †, and $^{-1}$ denote transposition, Hermitian transposition, complex conjugation, Moore-Penrose pseudoinverse, and matrix inversion respectively.

Kronecker product of two matrices $\boldsymbol{A}$, $\boldsymbol{B}$ is shown as $\boldsymbol{A} \otimes \boldsymbol{B}$ and the Khatri-Rao product (column-wise Kronecker product) as $\boldsymbol{A} \diamond \boldsymbol{B}$.

The matrix and the tensor norms are denoted as $||\cdot||_{\mathsf{X}}$, e.g., $||\cdot||_{\mathsf{E}}$, $||\cdot||_{\mathsf{F}}$ and $||\cdot||_{\mathsf{H}}$ are used for Euclidean norm for vectors, Frobenius norm for matrices, and higher-order norm for tensors, respectively. $E\{\cdot\}$ and $\mathrm{tr}\{\cdot\}$ are used to denote the expectation and the trace operators, respectively. $\mathrm{Tri}\{\cdot\}$ calculates the upper/lower triangular part of its argument and copies its Hermitian transpose to the other lower/upper triangular part. $\mathrm{diag}(\cdot)$ diagonalizes its argument column vector into a diagonal matrix and $\mathrm{vec}(\cdot)$ vectorizes the columns of its matrix argument. $\mathrm{Re}\{\cdot\}$, $\mathrm{Im}\{\cdot\}$, and $\arg\{\cdot\}$ are used to calculate the real part , imaginary part and the phase of a complex number, respectively. ∇ is the gradient operator and ∂ is used for partial derivative. $\mathrm{im}\,\{\cdot\}$, $\mathrm{null}\,\{\cdot\}$, $\mathrm{span}\,\{\cdot\}$ are used as image (column space), nullspace and span of their arguments, respectively.

2. TENSOR ALGEBRA AND HIGHER-ORDER SVD

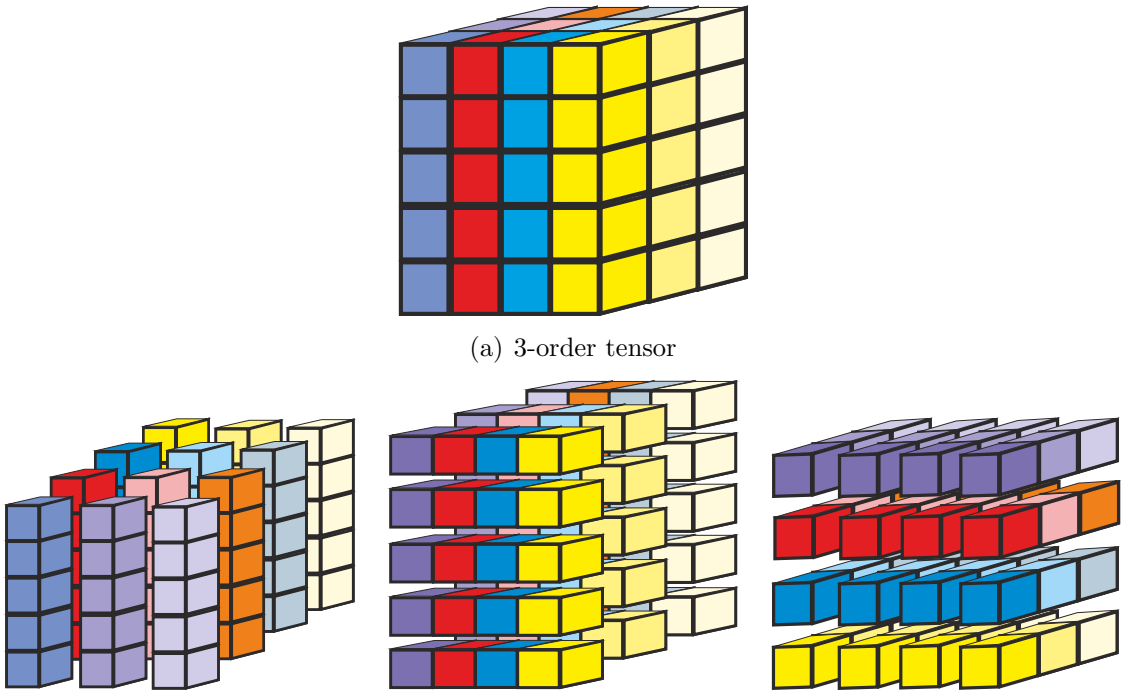
The concept of singular value decomposition (SVD) of matrices is one of the most fruitful developments in the world of linear algebra. The SVD concept has been widely used in the field of digital signal processing. More and more digital signal processing problems involve an appropriate manipulation for data that are structured in multidimensional arrays. Tensor algebra found its way as a promising tool for multidimensional arrays.

In this chapter, the existing framework of tensor algebra, High-order SVD (HOSVD) are presented. In addition, one application in using tensor-based subspace estimation in the digital signal processing field is given as well as the tensor-based subspace estimation via Kroncker structured projections.

2.1 Tensor Algebra

Tensor is a multi-dimensional array denoted as N-way array or N-order tensor. The N-order tensor is symbolized as $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ in the communities of numerical algebra and signal processing [dLdMV00]. A short summary of the used notation in this thesis is given similarly to standard notations in [dLdMV00] as following. The n-mode vectors is a generalization of row vectors and column vector of the tensor $\boldsymbol{\mathcal{A}}$ by varying the n-th index and fixing all other indices. And the n-mode subspace represents the vector space that spanned by the n-mode vectors. For instance, let us visualize a 3-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ and its n-mode vectors in a cube geometry as depicted in Fig. 2.1. Moreover, the n-mode unfolding of the tensor $\boldsymbol{\mathcal{A}}$ denoted a matrix $[\boldsymbol{\mathcal{A}}]_{(n)} \in \mathbb{C}^{I_n \times \frac{I}{I_n}}$ that contains all the n-mode vectors, where $I = \prod_{n=1}^{N} I_n,\ n = 1, 2, ..., N$. One example about ordering the n-mode unfolding columns is called Lathauwer's unfolding by using a reverse cyclical. In reverse cyclical, the ordering of n-mode unfolding columns started with the $(n+1)$-th index and proceeded backwards, up to $(n-1)$-th. Thus, the n-mode unfolding of the tensor $\boldsymbol{\mathcal{A}}$ can be given as $[\boldsymbol{\mathcal{A}}]_{(n)} \in \mathbb{C}^{I_n \times (I_{n+1} \cdot \ldots \cdot I_N \cdot I_1 \cdot \ldots \cdot I_{n-1})}$. Fig. 2.2 depicts the n-mode unfoldings of a 3-order tensor $\boldsymbol{\mathcal{A}}$ with $5 \times 4 \times 3$ dimensions, where

the indices i_1, i_2, i_3 vary in reverse cyclical way giving the unfolding matrices via lateral, vertical, and horizontal slices, respectively.



(a) 3-order tensor

(b) 1-mode vectors (c) 2-mode vectors (d) 3-mode vectors

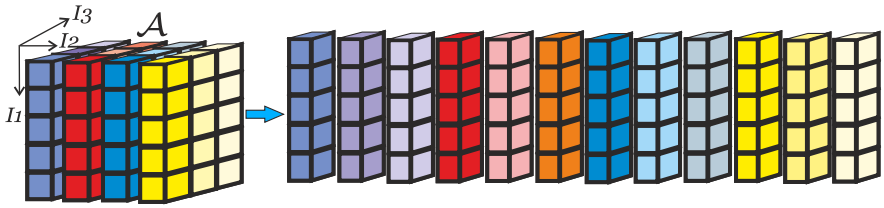
Fig. 2.1: 3-order tensor $\boldsymbol{\mathcal{A}}$ with $5 \times 4 \times 3$ dimensions and the n-mode vectors

2.1.1 Rank Properties of a N-Order Tensor

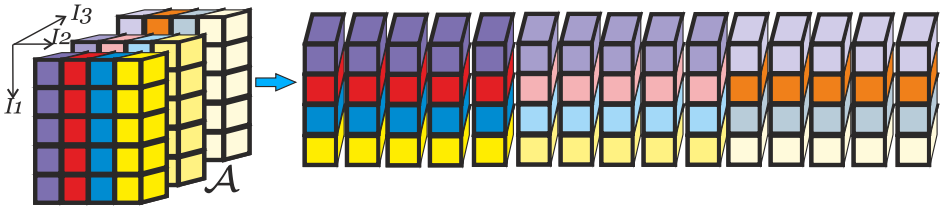
The rank of a N-order tensor differs compare to the matrix that has one rank. [dLdMV00] distinguished between two concepts of ranks, the n-rank of the n-mode unfolding, and the rank of the N-order tensor $\boldsymbol{\mathcal{A}}$. The n-rank of the n-mode unfolding has been denoted as n-rank $\{\boldsymbol{\mathcal{A}}\}$ = rank $\{[\boldsymbol{\mathcal{A}}]_{(n)}\}$. And the rank of N-order tensor has been defined as the minimal number of rank-1 tensors that yield $\boldsymbol{\mathcal{A}}$ in a linear combination; and has been represented as $R = \text{rank}\{\boldsymbol{\mathcal{A}}\}$. The rank-1 tensors can be represented by using the outer products of vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_N$ as

$$\boldsymbol{\mathcal{A}} = \boldsymbol{a}_1 \circ \boldsymbol{a}_2 \circ ... \circ \boldsymbol{a}_N, \tag{2.1}$$

with $\boldsymbol{a}_n \in \mathbb{C}^{I_n}$.



(a) 1-mode unfolding $[\mathcal{A}]_{(1)}$ with $5 \times (4 \cdot 3)$ dimensions



(b) 2-mode unfolding $[\mathcal{A}]_{(2)}$ with $4 \times (3 \cdot 5)$ dimensions

(c) 3-mode unfolding $[\mathcal{A}]_{(3)}$ with $3 \times (5 \cdot 4)$ dimensions

Fig. 2.2: n-mode unfolding of a 3-order tensor $\mathcal{A}$ with $5 \times 4 \times 3$ dimensions

2.1.2 Scalar Product, Orthogonality and Norm of Higher-Order Tensors

The scalar product $\langle \mathcal{A}, \mathcal{B} \rangle$ of two tensors $\mathcal{A}$ and $\mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ is the sum of the element-wise product of $\mathcal{A}$ and $\mathcal{B}^*$,

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_N=1}^{I_N} b^*_{i_1,i_2,\ldots,i_N} \cdot a_{i_1,i_2,\ldots,i_N}. \tag{2.2}$$

The orthogonality is defined as $\langle \mathcal{A}, \mathcal{B} \rangle = 0$ of two tensors $\mathcal{A}$ and $\mathcal{B} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$. In other words, the arrays of which the scalar product equals 0 are orthogonal as given in [dLdMV00]. The Frobenius-norm [dLdMV00],[KB09], or higher-order norm [HRDG08] of a tensor $\mathcal{A}$ is calculated using the square-root of the scalar product and defined as $||\mathcal{A}||_{\mathsf{H}} = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$.

2.1.3 Multiplication of a Tensor by a Matrix

The multiplication of a N-order tensor by a matrix is named as n-mode product [dLdMV00]. The n-mode product of a tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ and a matrix $\boldsymbol{U} \in$

$\mathbb{C}^{P_n \times I_n}$ yields a tensor $\boldsymbol{\mathcal{B}} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_{n-1} \times P_n \times I_{n+1} \times \ldots \times I_N}$ as

$$\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{A}} \times_n \boldsymbol{U}. \tag{2.3}$$

The n-mode product can be calculated by using the matrix multiplication between the matrix $\boldsymbol{U}$ and the n-mode unfolding $[\boldsymbol{\mathcal{A}}]_{(n)}$ giving the n-mode unfolding $[\boldsymbol{\mathcal{B}}]_{(n)}$ as

$$[\boldsymbol{\mathcal{B}}]_{(n)} = \boldsymbol{U} \cdot [\boldsymbol{\mathcal{A}}]_{(n)}. \tag{2.4}$$

For instance, Fig. 2.3 depicts the 1-mode product of a 3-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{4\times3\times2}$ with a matrix $\boldsymbol{U} \in \mathbb{C}^{5\times4}$, $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{A}} \times_1 \boldsymbol{U} \in \mathbb{C}^{5\times3\times2}$. The n-mode product satisfies two following

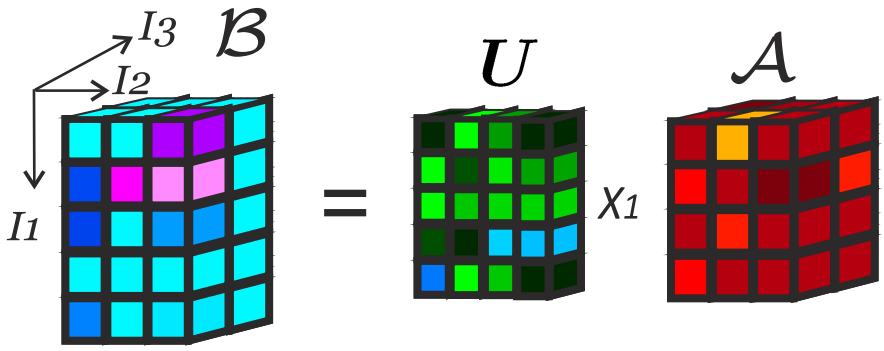


Fig. 2.3: 1-mode product of a 3-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{4\times3\times2}$ with a matrix $\boldsymbol{U} \in \mathbb{C}^{5\times4}$, $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{A}} \times_1 \boldsymbol{U} \in \mathbb{C}^{5\times3\times2}$

properties [dLdMV00]. The first one represents the commutativity property over the n-mode product as

$$\boldsymbol{\mathcal{A}} \times_m \boldsymbol{U} \times_n \boldsymbol{V} = \boldsymbol{\mathcal{A}} \times_n \boldsymbol{V} \times_m \boldsymbol{U}, \tag{2.5}$$

where $\boldsymbol{U} \in \mathbb{C}^{J_m \times I_m}$, $\boldsymbol{V} \in \mathbb{C}^{J_n \times I_n}$, and $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$. The second is a special case from the previous property at $(n = m)$. In other words, if $\boldsymbol{U}$, and $\boldsymbol{V} \in \mathbb{C}^{J_n \times I_n}$ have more than one n-mode product with the N-order tensor $\boldsymbol{\mathcal{A}}$, n-mode product between the N-order tensor and the total inverse inner product of matrices can be written as

$$\boldsymbol{\mathcal{A}} \times_n \boldsymbol{U} \times_n \boldsymbol{V} = \boldsymbol{\mathcal{A}} \times_n (\boldsymbol{V} \cdot \boldsymbol{U}). \tag{2.6}$$

2.1.4 Higher-Order SVD (HOSVD)

The SVD concept describes the relation between the column (row) vectors and the left (right) singular vectors of a matrix. The extension of SVD to HOSVD has been

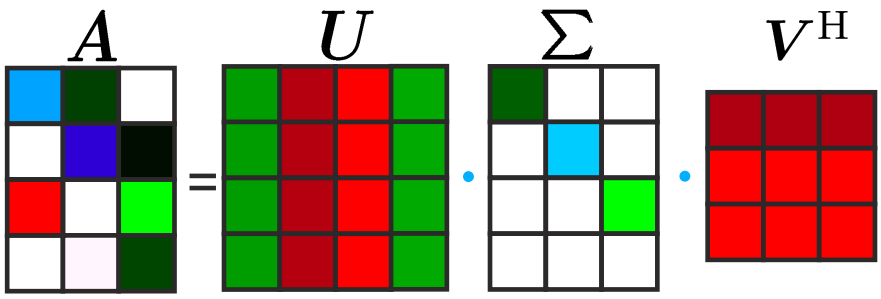


Fig. 2.4: SVD of a matrix $\boldsymbol{A} \in \mathbb{C}^{4\times3}$, where $\boldsymbol{U} \in \mathbb{C}^{4\times4}$, $\boldsymbol{V} \in \mathbb{C}^{3\times3}$ are unitary matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}^{4\times3}$ is a diagonal matrix

presented in [dLdMV00] depending on the Tucker model [Tuc66]. Tucker discussed the multilinear generalization of a three-way data $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1\times I_2\times I_3}$ and derived the tensor decomposition in a SVD terminology according to

$$a_{i_1 i_2 i_3} = \sum_{j_1}^{I_1}\sum_{j_2}^{I_2}\sum_{j_3}^{I_3} s_{j_1 j_2 j_3} u^{(1)}_{i_1 j_1} u^{(2)}_{i_2 j_2} u^{(3)}_{i_3 j_3}, \tag{2.7}$$

where, $s_{j_1 j_2 j_3}$ are elements of a tensor $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{I_1\times I_2\times I_3}$ with the property of all-orthogonality columns and $u_{i_1 j_1}$, $u_{i_2 j_2}$, $u_{i_3 j_3}$ are elements of unitary matrices. [dLdMV00] extended the decomposition for N-order tensor and used the term High-Order SVD (HOSVD). The HOSVD of a N-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1\times I_2\times\ldots\times I_N}$ in respect to the singular vectors of n-mode unfoldings $\boldsymbol{U}_n$, $n = 1, 2, ..., N$ can be written as

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{S}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 ... \times_N \boldsymbol{U}_N. \tag{2.8}$$

To generalize the HOSVD for the matrix case, De Lathauwer [dLdMV00] rewrote the matrix $\boldsymbol{A}$ in the 2-order tensor decomposition as

$$\boldsymbol{A} = \boldsymbol{U} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}^H = \boldsymbol{\Sigma} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}^* = \boldsymbol{\Sigma} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2, \tag{2.9}$$

where $\boldsymbol{U} \in \mathbb{C}^{I_1\times I_1}$, $\boldsymbol{V} \in \mathbb{C}^{I_2\times I_2}$ are unitary matrices, and $\boldsymbol{\Sigma} = \text{diag}\left(\sigma_1, \sigma_2, ..., \sigma_{\min(I_1,I_2)}\right)$ is a diagonal matrix consisting of singular values in descending order, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{\min(I_1,I_2)} \geq 0$. Fig. 2.4 visualizes the decomposition of matrix $\boldsymbol{A}$ according to the SVD concept.

Returning back to the N-order tensor in (2.8), the core tenor $\boldsymbol{\mathcal{S}}$ can be calculated according to

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{A}} \times_1 \boldsymbol{U}_1^H \times_2 \boldsymbol{U}_2^H ... \times_N \boldsymbol{U}_N^H, \tag{2.10}$$

Similarly to SVD, the core tensor $\boldsymbol{\mathcal{S}}$ has two properties [dLdMV00]:

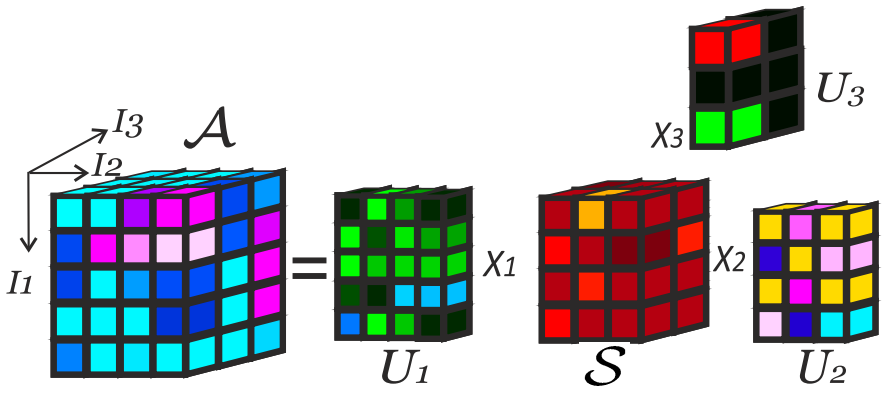


Fig. 2.5: HOSVD of a 3-order tensor $\boldsymbol{\mathcal{A}}$ of size $(5 \times 4 \times 3)$

1. all-orthogonality, the sub-tensors $\boldsymbol{\mathcal{S}}_{i_n=\alpha}$ and $\boldsymbol{\mathcal{S}}_{i_n=\beta}$, getting by fixing the i_n th index to α and β, are orthogonal for all possible values of n subject to $\alpha \neq \beta$

$$\langle \boldsymbol{\mathcal{S}}_{i_n=\alpha}, \boldsymbol{\mathcal{S}}_{i_n=\beta} \rangle = 0 \tag{2.11}$$

2. descending ordering of the Forbenius-norms,

$$||\boldsymbol{\mathcal{S}}_{i_n=1}||_{\mathsf{H}} \geq ||\boldsymbol{\mathcal{S}}_{i_n=2}||_{\mathsf{H}} \geq ... \geq ||\boldsymbol{\mathcal{S}}_{i_n=I_n}||_{\mathsf{H}} \geq 0. \tag{2.12}$$

The n-mode unfolding of the equation (2.8) has been expressed in [dLdMV00] as

$$[\boldsymbol{\mathcal{A}}]_{(n)} = \boldsymbol{U}_n \cdot [\boldsymbol{\mathcal{S}}]_{(n)} \cdot (\boldsymbol{U}_{n+1} \otimes \boldsymbol{U}_{n+2} \otimes ... \otimes \boldsymbol{U}_N \otimes \boldsymbol{U}_1 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_{n-1})^T, \tag{2.13}$$

where, $\boldsymbol{U}_n$ is calculated from the SVD of the n-mode unfolding of the tensor $\boldsymbol{\mathcal{A}}$ as

$$[\boldsymbol{\mathcal{A}}]_{(n)} = \boldsymbol{U}_n \cdot \boldsymbol{\Sigma}_n \cdot \boldsymbol{V}_n^H. \tag{2.14}$$

As we see before, $\boldsymbol{\Sigma}_n \in \mathbb{C}^{I_n \times I_n}$ is a diagonal matrix, $\boldsymbol{V}_n \in \mathbb{C}^{I_{n+1}I_{n+2}...I_N I_{N-1} I_{N-2}...I_{n-1} \times I_n}$ is the column-wise unitary matrix. Now, by substituting (2.14) into (2.13), we obtain

$$\boldsymbol{V}_n^H = \tilde{\boldsymbol{S}}_n \cdot (\boldsymbol{U}_{n+1} \otimes \boldsymbol{U}_{n+2} \otimes ... \otimes \boldsymbol{U}_N \otimes \boldsymbol{U}_1 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_{n-1})^T, \tag{2.15}$$

where $\tilde{\boldsymbol{S}}_n$ is a unitary matrix that normalized $[\boldsymbol{\mathcal{S}}]_{(n)}$ to unit-norm as

$$[\boldsymbol{\mathcal{S}}]_{(n)} = \boldsymbol{\Sigma}_n \cdot \tilde{\boldsymbol{S}}_n. \tag{2.16}$$

The n-mode unfolding of the core tensor $[\boldsymbol{\mathcal{S}}]_{(n)}$ can be expressed from each of the (2.14)

and (2.13), respectively as

$$[\boldsymbol{\mathcal{S}}]_{(n)} = \boldsymbol{U}_n^H \cdot [\boldsymbol{\mathcal{A}}]_{(n)} \cdot (\boldsymbol{U}_{n+1} \otimes \boldsymbol{U}_{n+2} \otimes ... \otimes \boldsymbol{U}_N \otimes \boldsymbol{U}_1 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_{n-1}), \quad (2.17)$$

$$[\boldsymbol{\mathcal{S}}]_{(n)} = \boldsymbol{\Sigma}_n \cdot \boldsymbol{V}_{(n)}^H \cdot (\boldsymbol{U}_{n+1} \otimes \boldsymbol{U}_{n+2} \otimes ... \otimes \boldsymbol{U}_N \otimes \boldsymbol{U}_1 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_{n-1}). \quad (2.18)$$

2.1.5 Low-Rank Approximation Property

The approximation of one matrix by another of lower rank have been illustrated in [EY36].The structure of the low-rank approximation is given in a term of signal subsapces and nullsubaces according to [Str93]. In fact, Strang defined the relation between the SVD of a matrix $\boldsymbol{A} \in \mathbb{C}^{I_1 \times I_2}$ and four fundamental subspaces such as:

1. the left singular vectors are the unitary basis for the column space and the left nullspace of $\boldsymbol{A}$, and
2. the right singular vectors are the unitary basis for the row space and the nullspace of $\boldsymbol{A}$.

Then, the SVD of $\boldsymbol{A}$ can be explicated in a form of signal subspaces and null spaces such as

$$\boldsymbol{A} = [\boldsymbol{U}_s \; \boldsymbol{U}_n] \cdot \begin{bmatrix} \boldsymbol{\Sigma}_s & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_n \end{bmatrix} \cdot [\boldsymbol{V}_s \; \boldsymbol{V}_n]^H, \quad (2.19)$$

where $\boldsymbol{U}_s \in \mathbb{C}^{I_1 \times d}$ and $\boldsymbol{U}_n \in \mathbb{C}^{I_1 \times (I_1 - d)}$ span the left singular vectors and the null spaces of the matrix $\boldsymbol{A}$, respectively. $\boldsymbol{\Sigma}_s \in \mathbb{C}^{d \times d}$ is a diagonal matrix contained the singular values of $\boldsymbol{A}$ and $\boldsymbol{\Sigma}_n \in \mathbb{C}^{(I_1 - d) \times (I_2 - d)}$ contains the noise values. According to the relation between SVD and the four fundamental subspaces, [HRDG08] wrote the SVD of a matrix $\boldsymbol{A}$ in low-rank approximation as

$$\boldsymbol{A} \approx \boldsymbol{U}_s \cdot \boldsymbol{\Sigma}_s \cdot \boldsymbol{V}_s^H \quad (2.20)$$

The truncated version of a matrix $\boldsymbol{A}$ with respect to the signal subspaces is depicted in Fig. 2.6. Since, the signal subspaces and nullspaces of a r-mode unfolding can be related via

$$\begin{aligned} \text{im} \left\{ [\boldsymbol{\mathcal{A}}]_{(r)} \right\} &\approx \text{im} \left\{ \boldsymbol{U}_r^{[s]} \right\}, \\ \text{null} \left\{ [\boldsymbol{\mathcal{A}}]_{(r)} \right\} &\approx \text{im} \left\{ \boldsymbol{U}_r^{[n]} \right\}, \end{aligned} \quad (2.21)$$

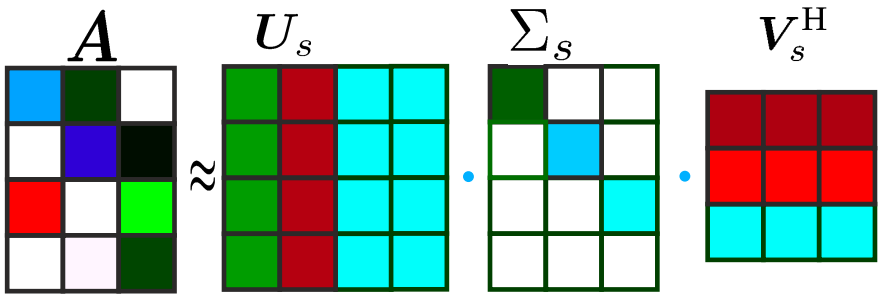


Fig. 2.6: Low-rank (Truncated) SVD of a matrix $\boldsymbol{A}$ of size (4×3)

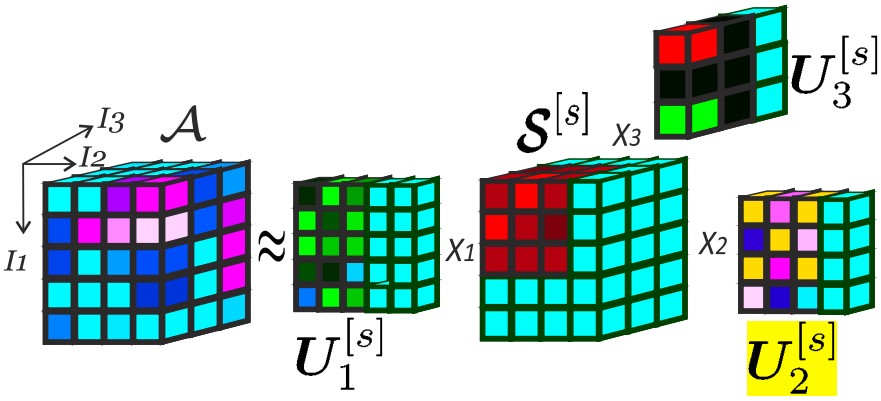


Fig. 2.7: Low-rank (Truncated) HOSVD of a tensor $\boldsymbol{\mathcal{A}}$ of size $(5 \times 4 \times 3)$

HOSVD of a R-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_R}$ can be written as

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{S}} \times_1 \left[\boldsymbol{U}_1^{[s]}\ \boldsymbol{U}_1^{[n]}\right] \times_2 \left[\boldsymbol{U}_2^{[s]}\ \boldsymbol{U}_2^{[n]}\right] \ldots \times_{R+1} \left[\boldsymbol{U}_{R+1}^{[s]}\ \boldsymbol{U}_{R+1}^{[n]}\right], \tag{2.22}$$

where $\boldsymbol{U}_r^{[s]}$ and $\boldsymbol{U}_r^{[n]}$ span the signal subspaces and nullspaces r-mode unfolding, respectively. Then, the low-rank approximation of $\boldsymbol{\mathcal{A}}$ is computed in similar way to truncated SVD as in [HRDG08] according to

$$\boldsymbol{\mathcal{A}} \approx \boldsymbol{\mathcal{S}}^{[s]} \times_1 \boldsymbol{U}_1^{[s]} \times_2 \boldsymbol{U}_2^{[s]} \ldots \times_R \boldsymbol{U}_R^{[s]}. \tag{2.23}$$

Fig. 2.7 visualizes the truncated HOSVD concept for a 3-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{5\times4\times3}$, where $\boldsymbol{\mathcal{S}}^{[s]} \in \mathbb{C}^{5\times4\times3}$, $\boldsymbol{U}_1^{[s]} \in \mathbb{C}^{5\times3}$, $\boldsymbol{U}_2^{[s]} \in \mathbb{C}^{4\times3}$, $\boldsymbol{U}_3^{[s]} \in \mathbb{C}^{3\times2}$. It's worth to say that the low-rank approximation reduces the mathematical complexity with a good approximation [HRDG08]. In the following, we are going to use this property in the tensor-based subspace estimation.

2.2 Tensor-Based Subspace Estimation

Tensor based-subspace estimation has been applied to estimate the signal parameters of multidimensional harmonic retrieval data model as shown in [HRDG08]. More manipulation has been developed to reduce the computational complexity in [RBHW09b] and [CH13] for the tensor-based subspace tracking via Kronecker structured projections without the need to calculate the core tensor of the data.

2.2.1 Data Model for a Multidimensional Harmonic Retrieval Problem

For multidimensional harmonic retrieval data model, we have d sources of narrowband noncoherent plane wavefronts and array of M senors on a far field criteria. Thus, the data model has been assumed to has a uniform sampling in the spatial domain and the goal is to estimate the signal parameters, i.e., Direction of Arrival (AOD). The data model is given in a matrix approach as well as a tensor approach as introduced in [HRDG08]. In the following, we are going to summarize both of the approaches and estimate the tensor-based signal subspace.

- **Matrix Based Data Model**

  Let us assume that a data vector $\boldsymbol{x} \in \mathbb{C}^M$ has been observed at M sensors which have a uniform antenna array; and thus composed of d narrow-band signal waves $\boldsymbol{s} \in \mathbb{C}^d$ superimposed by an additive noise vector $\boldsymbol{n} \in \mathbb{C}^M$. Then, the data are rearranged in a matrix form at each time slot without taking to account the multidimensional structure of the antenna array as

  $$\boldsymbol{X} = \boldsymbol{A} \cdot \boldsymbol{S} + \boldsymbol{N} \in \mathbb{C}^{M \times N}, \tag{2.24}$$

  where $\boldsymbol{A} = [\boldsymbol{a}(\omega_1), ..., \boldsymbol{a}(\omega_d)] \in \mathbb{C}^{M \times d}$ is the array steering matrix corresponding to the direction of arrival $\boldsymbol{a}(\omega_i) = \left[1, e^{j\omega_i}, ..., e^{j(M-1)\omega_i}\right]$, where $\omega_i,\ i = 1, 2, ..., d$ are the unknown spatial frequencies.

  And so that for R-dimensional signals, the array steering matrix $\boldsymbol{A}$ can be given in a term of mixing matrices $\boldsymbol{A}_r$ as

  $$\boldsymbol{A} = \boldsymbol{A}_1 \diamond \boldsymbol{A}_2 \diamond ... \diamond \boldsymbol{A}_R \in \mathbb{C}^{M \times d},\ r = 1, 2, ..., R, \tag{2.25}$$

  where $\boldsymbol{A}_r = [\boldsymbol{a}_r(\omega_{r_1}), ..., \boldsymbol{a}_r(\omega_{r_d})] \in \mathbb{C}^{M_r \times d}$ is the mixing matrix in the r-th mode, $M = \prod_{r=1}^{R} M_r$ as introduced in [HRDG08]. The array steering vectors can be

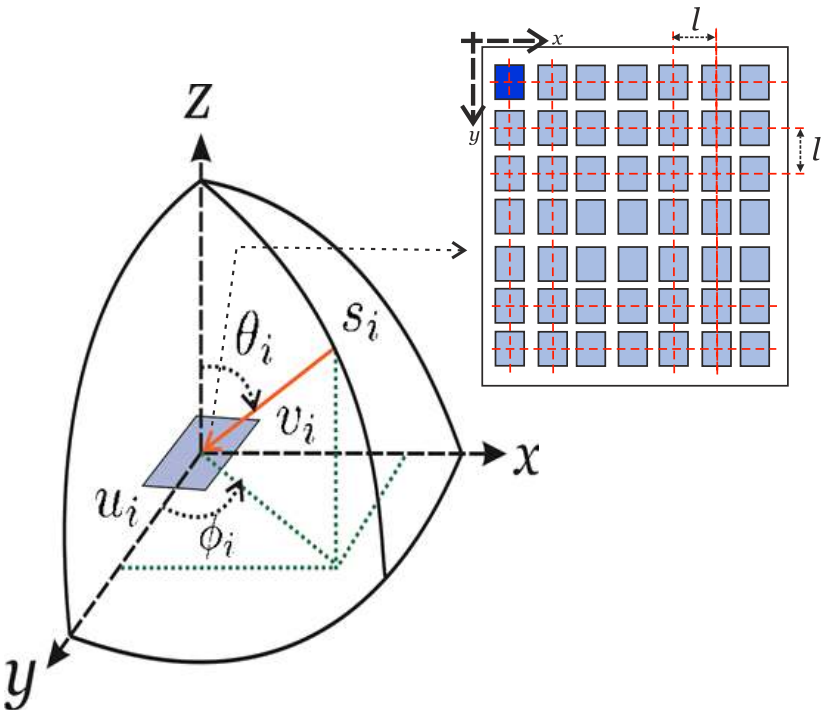


Fig. 2.8: A three dimensional Cartesian coordinate system contains the projections of a i-th point source s_i on the euclidean plane, μ_i and ν_i. The azimuth and elevation angles are $-180 \leq \phi_i \leq 180$ and $0 \leq \theta_i \leq 90$, respectively.

written such as

$$\boldsymbol{a}_r(\omega_{r_i}) = \left[1, e^{j\omega_{r_i}}, ..., e^{j(M_r-1)\omega_{r_i}}\right], \tag{2.26}$$

where ω_{r_i} is the i-th spatial frequency on r-th dimension. For instance, if the antenna array is a Uniform rectangular array (URA), we can define

$$\begin{aligned} \omega_{1_i} = \mu_i = 2\pi \frac{l}{\lambda} u_i, \;\; u_i &= \frac{1}{2} \cos\phi_i \sin\theta_i \\ \omega_{2_i} = \nu_i = 2\pi \frac{l}{\lambda} v_i, \;\; v_i &= \frac{1}{2} \sin\phi_i \sin\theta_i, \end{aligned} \tag{2.27}$$

where l is the distance between the neighboring sensors, λ is the wavelength, and ϕ_i and θ_i are the azimuth and the elevation angles on the Cartesian coordinate system, respectively. Fig. 2.8 visualizes a URA centro-symmetric sensors array of $M = 7 \times 7$ elements lying in the $x - y$ plane.

- **Tensor Based Data Model**

  In multidimensional harmonic retrieval, the tensor data model $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M_1 \times M_2 \times ... \times M_R \times N}$ is reformulated according to [HRDG08] as

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{A}} \times_{R+1} \boldsymbol{S} + \boldsymbol{\mathcal{N}}, \tag{2.28}$$

where $\boldsymbol{S} \in \mathbb{C}^{d \times N}$ contains d narrowband signals same as in (2.24), $\boldsymbol{\mathcal{N}}$ composes the noise samples with the same size of the observations data tensor $\boldsymbol{\mathcal{X}}$, and $\boldsymbol{\mathcal{A}}$ is a array steering tensor with $M_1 \times M_2 \times ... \times M_R \times d$ dimensions.

As we see above, the tensor data model kept on the multidimensional structure of the antenna array compared to the matrix data model. And to get the link between both of them, let us define the array steering tensor in the form of r-mode mixing matrices according to [HRDG08] as

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{I}}_{R+1,d} \times_1 \boldsymbol{A}_1 ... \times_R \boldsymbol{A}_R \in \mathbb{C}^{M_1 \times M_2 \times ... \times M_R \times d} \tag{2.29}$$

where $\boldsymbol{\mathcal{I}}_{R+1,d}$ is the $(R+1)$-dimensional identity tensor of size $d \times d \times ... \times d$ such that

$$(\boldsymbol{\mathcal{I}}_{R+1,d})_{i_1,i_2,...,i_{R+1}} = \begin{cases} 1 & \text{if } i_1 = i_2 = ... = i_{R+1}, \\ 0 & \text{otherwise.} \end{cases}$$

Then, the connection between the matrix and the tensor data models is given by the transpose of the unfoldings as

$$\begin{aligned} \boldsymbol{X} &= [\boldsymbol{\mathcal{X}}]^T_{(R+1)} \\ \boldsymbol{A} &= [\boldsymbol{\mathcal{A}}]^T_{(R+1)} \\ \boldsymbol{N} &= [\boldsymbol{\mathcal{N}}]^T_{(R+1)} \end{aligned} \tag{2.30}$$

2.2.2 HOSVD-Based Signal Subspace Estimation

The derivation of HOSVD-based signal subspace estimation is clarified in [HRDG08]. At first, the HOSVD of the observation tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times I_2 \times .... \times I_{R+1}}$ is given in the form of signal subspace and null subspace as the following

$$\boldsymbol{\mathcal{X}} = \hat{\boldsymbol{\mathcal{S}}} \times_1 \left[\hat{\boldsymbol{U}}_1^{[s]} \; \hat{\boldsymbol{U}}_1^{[n]}\right] \times_2 \left[\hat{\boldsymbol{U}}_2^{[s]} \; \hat{\boldsymbol{U}}_2^{[n]}\right] ... \times_{R+1} \left[\hat{\boldsymbol{U}}_{R+1}^{[s]} \; \hat{\boldsymbol{U}}_{R+1}^{[n]}\right], \tag{2.31}$$

where $\hat{\boldsymbol{U}}_r^{[s]} \in \mathbb{C}^{M_r \times p_r}$ and $\hat{\boldsymbol{U}}_r^{[n]} \in \mathbb{C}^{M_r \times (M_r - p_r)}$ for $r = 1, 2, ..., R+1$. Then, the tensor $\boldsymbol{\mathcal{X}}$ has been approximated to the truncated HOSVD as

$$\boldsymbol{\mathcal{X}} \approx \hat{\boldsymbol{\mathcal{S}}}^{[s]} \times_1 \hat{\boldsymbol{U}}_1^{[s]} \times_2 \hat{\boldsymbol{U}}_2^{[s]} ... \times_{R+1} \hat{\boldsymbol{U}}_{R+1}^{[s]}, \tag{2.32}$$

where the $\hat{\boldsymbol{\mathcal{S}}}^{[s]} \in \mathbb{C}^{p_1 \times p_2 \times \ldots p_R \times d}$ is the truncated core tensor. Similar to (2.10), we could write the truncated core tensor $\hat{\boldsymbol{\mathcal{S}}}^{[s]}$ in the HOSVD form as shown in

$$\hat{\boldsymbol{\mathcal{S}}}^{[s]} = \boldsymbol{\mathcal{X}} \times_1 \hat{\boldsymbol{U}}_1^{[s]^H} \times_2 \hat{\boldsymbol{U}}_2^{[s]^H} \ldots \times_{R+1} \hat{\boldsymbol{U}}_{R+1}^{[s]^H}. \tag{2.33}$$

The HOSVD tensor-based signal subspace can be described in the term of core tensor and multidimensional extension of the basis $\hat{\boldsymbol{U}}_{[s]} \in \mathbb{C}^{M_1 \times M_2 \times \ldots \times M_R \times d}$ according to [HRDG08] as

$$\hat{\boldsymbol{\mathcal{U}}}^{[s]} = \hat{\boldsymbol{\mathcal{S}}}^{[s]} \times_1 \hat{\boldsymbol{U}}_1^{[s]} \times_2 \hat{\boldsymbol{U}}_2^{[s]} \ldots \times_R \hat{\boldsymbol{U}}_R^{[s]}. \tag{2.34}$$

Compare to low-rank approximation of SVD, the observation data matrix in (2.24) can be approximated as

$$\boldsymbol{X} = \boldsymbol{U} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}^{\mathrm{H}} \;\; \approx \;\; \boldsymbol{U}_s \cdot \boldsymbol{\Sigma}_s \cdot \boldsymbol{V}_s^{\mathrm{H}}. \tag{2.35}$$

Since the approximation for the matrix and tensor cases yield $\mathrm{im}\{\boldsymbol{U}_s\} = \mathrm{im}\{\boldsymbol{A}\}$, we can define the link between the matrix-based and tensor-based signal subspaces based on the relation in (2.30). [HRDG08] clarified that the column spaces of the r-mode unfolding of the array steering tensor $[\boldsymbol{\mathcal{A}}]_{(r)}$ span the r-mode vector spaces form such as

$$\mathrm{im}\left\{[\boldsymbol{\mathcal{A}}]_{(R+1)}^T\right\} \;\; \approx \;\; \mathrm{im}\left\{\left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]_{(R+1)}^T\right\}. \tag{2.36}$$

We assume in our study that the number of wavefronts d is known and equals to the rank of the measurement matrix and the rank of the measurement tensor, respectively. Moreover, the equations (2.32) and (2.35) hold as good as in the noiseless case [HRDG08]. By using the property (2.5) and substituting each of (2.35) and (2.32) in (2.32) we obtain

$$\begin{aligned} \boldsymbol{U}_s \cdot \boldsymbol{\Sigma}_s \cdot \boldsymbol{V}_s^{\mathrm{H}} \;\; &\approx \;\; \left[\hat{\boldsymbol{\mathcal{U}}}^{[s]} \times_{R+1} \hat{\boldsymbol{U}}_{R+1}^{[s]}\right]_{(R+1)}^T \\ \boldsymbol{U}_s \cdot \boldsymbol{T}_s \;\; &\approx \;\; \left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]_{(R+1)}^T \cdot (\hat{\boldsymbol{U}}_{R+1}^{[s]})^T, \end{aligned} \tag{2.37}$$

where $\boldsymbol{T}_s = \boldsymbol{\Sigma}_s \cdot \boldsymbol{V}_s^{\mathrm{H}}$. In other words, the tensor-based signal subspace span the same signal subspace as shown in

$$\mathrm{im}\left\{\hat{\boldsymbol{U}}_s\right\} \approx \mathrm{im}\left\{\left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]_{(R+1)}^T\right\}. \tag{2.38}$$

Right now, the tensor-based signal subspace can be calculated by computing the core tensor of the observation data tensor $\boldsymbol{\mathcal{X}}$. More manipulation has been applied in [RBHW09a] in order to simplify the computation by avoiding the tensor calculation as following: The n-mode unfolding of the observation data tensor $\boldsymbol{\mathcal{X}}$ has been calculated as

$$[\boldsymbol{\mathcal{X}}]^T_{(R+1)} \approx \left(\hat{\boldsymbol{U}}_1^{[s]} \otimes \hat{\boldsymbol{U}}_2^{[s]} \otimes ... \otimes \hat{\boldsymbol{U}}_R^{[s]}\right) \cdot \left[\hat{\boldsymbol{\mathcal{S}}}^{[s]}\right]^T_{(R+1)} \cdot \left(\hat{\boldsymbol{U}}_{R+1}^{[s]}\right)^T. \tag{2.39}$$

To abstract (2.39) more, a unitary matrix $\tilde{\boldsymbol{S}}_{R+1}^{[s]} \in \mathbb{C}^{d \times \prod_{r=1}^{R} p_r}$ has been defined according to (2.16) as

$$[\boldsymbol{\mathcal{S}}]_{(n)} = \boldsymbol{\Sigma}_n \cdot \tilde{\boldsymbol{S}}_n. \tag{2.40}$$

Now, the substituting (2.40) into (2.39) and also considering the link in the equations (2.13), (2.14) and (2.15) yield

$$\begin{aligned} [\boldsymbol{\mathcal{X}}]^T_{(R+1)} &\approx \left(\hat{\boldsymbol{U}}_1^{[s]} \otimes \hat{\boldsymbol{U}}_2^{[s]} \otimes ... \otimes \hat{\boldsymbol{U}}_R^{[s]}\right) \cdot \left(\tilde{\boldsymbol{S}}_{R+1}^{[s]}\right)^T \cdot \hat{\boldsymbol{\Sigma}}_{R+1}^{[s]} \cdot \left(\hat{\boldsymbol{U}}_{R+1}^{[s]}\right)^T \\ &\approx \hat{\boldsymbol{U}}_s \cdot \hat{\boldsymbol{\Sigma}}_{R+1}^{[s]} \cdot \left(\hat{\boldsymbol{U}}_{R+1}^{[s]}\right)^T. \end{aligned} \tag{2.41}$$

Linking to the SVD concept, the signal subspace of $\boldsymbol{X}$ can be written using the truncated SVD as

$$\hat{\boldsymbol{U}}_s = \boldsymbol{X} \cdot \hat{\boldsymbol{V}}_s \cdot \hat{\boldsymbol{\Sigma}}_s^{-1}. \tag{2.42}$$

By inserting (2.42) into (2.41), the signal subspace $\hat{\boldsymbol{U}}_s$ can be calculated with respect to the unfolding date tenor as

$$[\boldsymbol{\mathcal{X}}]^T_{(R+1)} \cdot \hat{\boldsymbol{U}}_{R+1}^{[s]^*} \cdot \hat{\boldsymbol{\Sigma}}_s^{-1} = \boldsymbol{X} \cdot \hat{\boldsymbol{V}}_s \cdot \hat{\boldsymbol{\Sigma}}_s^{-1} = \hat{\boldsymbol{U}}_s. \tag{2.43}$$

Consequently, by calculating the $(R+1)$-mode unfolding of (2.34) and applying the transpose operator, the $R+1$-mode unfolding of the tensor subspace $\hat{\boldsymbol{\mathcal{U}}}$ can be expressed as

$$\left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]^T_{(R+1)} = \left(\hat{\boldsymbol{U}}_1^{[s]} \otimes \hat{\boldsymbol{U}}_2^{[s]} \otimes ... \otimes \hat{\boldsymbol{U}}_R^{[s]}\right) \cdot \left[\hat{\boldsymbol{\mathcal{S}}}^{[s]}\right]^T_{(R+1)}. \tag{2.44}$$

Similarly, by substituting the transpose equation (2.40) into (2.44), we can obtain the following relation

$$\begin{aligned} \left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]^T_{(R+1)} &= \left(\hat{\boldsymbol{U}}_1^{[s]} \otimes \hat{\boldsymbol{U}}_2^{[s]} \otimes ... \otimes \hat{\boldsymbol{U}}_R^{[s]}\right) \cdot \left(\tilde{\boldsymbol{S}}_{R+1}^{[s]}\right)^T \cdot \hat{\boldsymbol{\Sigma}}_{R+1}^{[s]} \\ &\approx \hat{\boldsymbol{U}}_s \cdot \hat{\boldsymbol{\Sigma}}_{R+1}^{[s]}. \end{aligned} \tag{2.45}$$

In [HRDG08], the tensor-based subspace estimate $\left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]^T_{(R+1)}$ outperforms the matrix-based subspace estimate $\hat{\boldsymbol{U}}_s$ due to the structured denoising in each mode $r = 1, 2, ..., R$ with $d < M_r$. In addition, the [HRDG08] mentioned that we can calculate the tensor-based signal subspace without calculating the core tensor reducing the mathematical complexity one step more.

2.2.3 Structured Projections for HOSVD-Based Signal Subspace Estimation

The tensor-based signal subspace estimation $\hat{\boldsymbol{\mathcal{U}}}^{[s]}$ has been written for a notational convenience by normalizing (2.34) via $\hat{\boldsymbol{\Sigma}}_s^{-1}$ according to [RBHW09a] such as

$$\hat{\boldsymbol{\mathcal{U}}}^{[s]} = \hat{\boldsymbol{\mathcal{S}}}^{[s]} \times_1 \hat{\boldsymbol{U}}_1^{[s]} \times_2 \hat{\boldsymbol{U}}_2^{[s]} ... \times_R \hat{\boldsymbol{U}}_R^{[s]} \times_{R+1} \hat{\boldsymbol{\Sigma}}_s^{-1}, \tag{2.46}$$

where $\hat{\boldsymbol{\Sigma}}_s = \hat{\boldsymbol{\Sigma}}_{R+1}^{[s]}$. More manipulation has been performed in [CH13] in order to reduce the mathematical complexity as shown in the following. As aforementioned, the tensor subspace can be calculated without computing the core tensor by inserting the core tensor $\hat{\boldsymbol{\mathcal{S}}}^{[s]}$ into (2.46) and using the properties in (2.5). Wherefore, the tensor-based signal subspace estimation can be clarified according to

$$\hat{\boldsymbol{\mathcal{U}}}^{[s]} = \boldsymbol{\mathcal{X}} \times_1 \hat{\boldsymbol{T}}_1 \times_2 \hat{\boldsymbol{T}}_2 ... \times_R \hat{\boldsymbol{T}}_R \times_{R+1} \left(\hat{\boldsymbol{\Sigma}}_s^{-1} \cdot \hat{\boldsymbol{U}}_{R+1}^{[s]^H}\right) \tag{2.47}$$

where $\hat{\boldsymbol{T}}_r = \hat{\boldsymbol{U}}_r^{[s]} \cdot \hat{\boldsymbol{U}}_r^{[s]^H}$ is the projection matrix onto the space spanned by the r-mode vectors.

Using the property (2.13), the transpose of the $(R+1)$-mode unfolding of the tensor subspace $\left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]^T_{(R+1)}$ is calculated via [RBHW09b] as

$$\left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]^T_{(R+1)} = \left(\hat{\boldsymbol{T}}_1 \otimes \hat{\boldsymbol{T}}_2 ... \otimes \hat{\boldsymbol{T}}_R\right) \cdot [\boldsymbol{\mathcal{X}}]^T_{(R+1)} \cdot \hat{\boldsymbol{U}}_{R+1}^{[s]^*} \cdot \hat{\boldsymbol{\Sigma}}_s^{-1}. \tag{2.48}$$

Consequently, by substituting (2.43) in (2.48), the HOSVD-based subspace estimate becomes the projection of the matrix-based subspace estimate onto Kronecker structure of the r-mode projectors as

$$\left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]^T_{(R+1)} = \left(\hat{\boldsymbol{T}}_1 \otimes \hat{\boldsymbol{T}}_2 ... \otimes \hat{\boldsymbol{T}}_R\right) \cdot \hat{\boldsymbol{U}}_s. \tag{2.49}$$

The complexity in (2.49) is $O(M^2 d)$ multiplications as shown in [RBHW09b] and [CH13]. Moreover, the [CH13] explicated that the complexity can be reduced by using

two of the Kronecker product properties stated as follows.

1. Given two pairs of matrices $\boldsymbol{A}, \boldsymbol{C}$ and $\boldsymbol{B}, \boldsymbol{D}$ of proper dimensions, the Kronecker product $(\boldsymbol{A} \cdot \boldsymbol{C}) \otimes (\boldsymbol{B} \cdot \boldsymbol{D})$ can be written as,

$$(\boldsymbol{A} \cdot \boldsymbol{C}) \otimes (\boldsymbol{B} \cdot \boldsymbol{D}) = (\boldsymbol{A} \otimes \boldsymbol{B}) \cdot (\boldsymbol{C} \otimes \boldsymbol{D}), \tag{2.50}$$

   which is called the mixed-product property of the Kronecker product.

2. Transposition and complex conjugation operations are distributive over the Kronecker product as

$$(\boldsymbol{A} \otimes \boldsymbol{B})^H = \left(\boldsymbol{A}^H \otimes \boldsymbol{B}^H\right) \tag{2.51}$$

Then the projectors in equation (2.49) have been calculated using (2.50) and (2.51) as

$$\begin{aligned} \left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]^T_{(R+1)} &= \left(\hat{\boldsymbol{U}}_1^{[s]} \cdot \hat{\boldsymbol{U}}_1^{[s]^H}\right) \otimes \left(\hat{\boldsymbol{U}}_2^{[s]} \cdot \hat{\boldsymbol{U}}_2^{[s]^H}\right) ... \otimes \left(\hat{\boldsymbol{U}}_R^{[s]} \cdot \hat{\boldsymbol{U}}_R^{[s]^H}\right) \cdot \hat{\boldsymbol{U}}_s \\ &= \left(\hat{\boldsymbol{U}}_1^{[s]} \otimes \hat{\boldsymbol{U}}_2^{[s]} ... \otimes \hat{\boldsymbol{U}}_R^{[s]}\right) \cdot \left(\hat{\boldsymbol{U}}_1^{[s]} \otimes \hat{\boldsymbol{U}}_2^{[s]} ... \otimes \hat{\boldsymbol{U}}_R^{[s]}\right)^H \cdot \hat{\boldsymbol{U}}_s. \end{aligned} \tag{2.52}$$

Note that the projectors $\hat{\boldsymbol{T}}_r$ is written again in the form $\hat{\boldsymbol{U}}_r^{[s]} \cdot \hat{\boldsymbol{U}}_r^{[s]^H}$. Defining a lower-dimensional space $\hat{\tilde{\boldsymbol{U}}}_s \in \mathbb{C}^{d^R \times d}$ which represents the subspace in result of the projection, the equation above becomes

$$\left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]^T_{(R+1)} = \left(\hat{\boldsymbol{U}}_1^{[s]} \otimes \hat{\boldsymbol{U}}_2^{[s]} ... \otimes \hat{\boldsymbol{U}}_R^{[s]}\right) \cdot \hat{\tilde{\boldsymbol{U}}}_s. \tag{2.53}$$

This gives a complexity in the order of $O(Md^{(R+1)})$, thus linear in M.

Next we will incorporate the Forward Backward Averaging (FBA) concept to write the "normal"-and "mixed-product Kronecker" structured HOSVD-based subspace estimation, according to (2.52) and (2.53), respectively, in a form of real-valued processing.

3. TENSOR-BASED SUBSPACE TRACKING VIA STRUCTURED PROJECTIONS

Tracking the signal subspace plays an important role in the field of digital signal processing for various applications, e.g., signal parameters tracking. In this area, a lot of tracking algorithms have been developed such as Projection Approximation Subspace Tracking (PAST) [Yan95].The PAST algorithm has been applied for tracking and updating signal subspaces that slowly change.

Recently, this algorithm has been modified for tensor-based subspace tracking as introduced in [CH13]. Here, a generic framework via "normal"-and "mixed-product Kronecker" structured HOSVD projections is used based on an extension version of the PAST algorithm. Our contribution in this regard is to incorporate the Forward Backward Averaging (FBA) concept which we expect to improve the tracking algorithm as well as to decrease the mathematical complexity.

In this chapter, we give a brief summary about the PAST and the Extended PAST algorithms. Then, we propose the FBA-PAST algorithm based on incorporate the FBA processing and extend it again to Extended FBA-PAST based on tensor-based subspace tracking via structured projections.

3.1 Projection Approximation Subspace Tracking (PAST)

The PAST algorithm is an adaptive algorithm that updates the signal subspace recursively, as shown in [Yan95]. Yang utilized the projection approximation approach to simplify its unconstrained minimization problem by using the Recursive Least Squares (RLS) technique.

To clarify this algorithm, we consider a data model of a multidimensional harmonic retrieval problem to be given according to (2.24). So that, an observed vector $\boldsymbol{x}(n) \in \mathbb{C}^M$ at n-th snapshot can be expressed as

$$\boldsymbol{x}(n) = \boldsymbol{A}\boldsymbol{s}(n) + \boldsymbol{w}(n), \tag{3.1}$$

where $\boldsymbol{A} \in \mathbb{C}^{M \times d}$ denotes a deterministic array steering matrix, $\boldsymbol{s}(n) = [s_1(n), ..., s_d(n)]^T \in \mathbb{C}^d$ symbolizes a random sources vector with a non-singular correlation matrix $\boldsymbol{C}_{ss} = E[\boldsymbol{s}(n)\boldsymbol{s}(n)^H]$, and $\boldsymbol{w}(n)$ is an additive noise vector with the same size of the observation data vector.

The covariance matrix of the observed data $\boldsymbol{C}_{xx} \in \mathbb{C}^{M \times M}$ is given as

$$\boldsymbol{C}_{xx} = E[\boldsymbol{x}(n)\boldsymbol{x}(n)^H] = \boldsymbol{A}\boldsymbol{C}_{ss}\boldsymbol{A}^H + \sigma_{ww}^2 \boldsymbol{I}_M, \tag{3.2}$$

where σ_{ww}^2 is the noise variance. Related to the additive noise, it can be seen that if $\sigma_{ww}^2 = 0$, the columns of the array steering matrix $\boldsymbol{A}$ span exactly the d-dimensional signal subspace $\boldsymbol{U}_s$ of the observed data. However, in the case when the observation data is interrupt with an additive noise, the equality becomes accepted for a wide observation window, i.e., $N \to \infty$ yielding

$$\mathrm{im}\{\boldsymbol{A}\} = \mathrm{im}\{\boldsymbol{U}_s\}. \tag{3.3}$$

In order to explain the recursive method, we visualizes the RLS filter in Fig. 3.1, where $\hat{\boldsymbol{x}}(n)$ denotes the estimate of $\boldsymbol{x}(n)$, $\boldsymbol{e}(n)$ symbolizes the error between the estimated signal and the current observation, and $\Delta_{\boldsymbol{U}_n}$ is the correction factor.

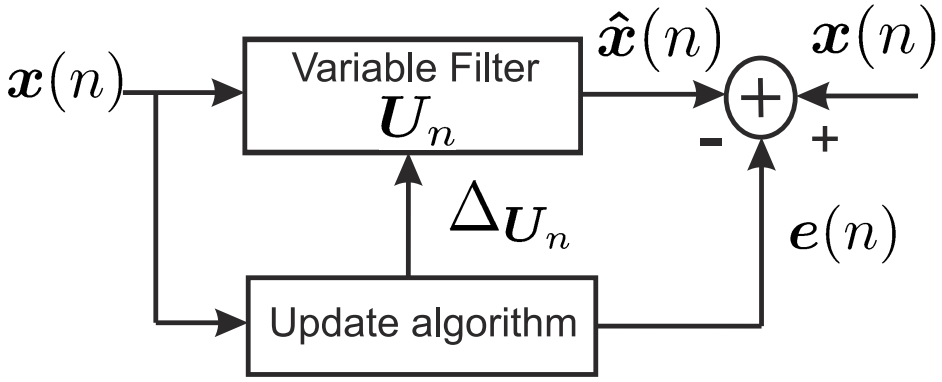


Fig. 3.1: RLS filter. $\boldsymbol{x}(n)$ represents the observation data vector at n-th snapshot, $\hat{\boldsymbol{x}}(n)$ denotes the estimated one of $\boldsymbol{x}(n)$, $\boldsymbol{e}(n)$ symbolizes the error between the estimated signal and the current observation, and $\Delta_{\boldsymbol{U}_n}$ is the correction factor.

Now, we can write the cost function of the signal subspace, according to [Yan95], as follows

$$J(\boldsymbol{U}_s(n)) = \sum_{i=1}^{N} \beta^{n-i} \parallel \boldsymbol{x}(i) - \boldsymbol{U}_s(n) \cdot \boldsymbol{y}(i) \parallel^2, \tag{3.4}$$

where $\boldsymbol{y}(i) = \boldsymbol{U}_s^H(n-1) \cdot \boldsymbol{x}(i) \in \mathbb{C}^d$ represents a projection matrix and β is a forgetting factor in the range $0 \leq \beta \leq 1$.

According to [Yan95], we discuss the impact of the forgetting factor β on the cost function. Since the data in the distance past are down-weighted by using the β, the tracking capability is still available when the system operates in a non-stationary environment i.e., the array steering $\boldsymbol{A}$ is time-variant. Therefore, the effective window length for $\beta < 1$ has been considered as $1/(1-\beta)$ when $n \gg 1$.

The cost function has been written again in [Yan95] by using the trace operator of correlation matrices such as

$$\begin{aligned} \boldsymbol{J}(\boldsymbol{U}_s(n)) &= \mathrm{tr}[\boldsymbol{C}_{xx}(n)] + \mathrm{tr}[\boldsymbol{U}_s(n) \cdot \boldsymbol{C}_{yx}(n)] \\ &+ \mathrm{tr}[\boldsymbol{C}_{xy}(n) \cdot \boldsymbol{U}_s^H(n)] + \mathrm{tr}[\boldsymbol{U}_s(n) \cdot \boldsymbol{C}_{yy}(n) \cdot \boldsymbol{U}_s^H(n)], \end{aligned} \tag{3.5}$$

where the correlation matrices can be given by

$$\boldsymbol{C}_{xx}(n) = \sum_{i=1}^{N} \beta^{n-i} \boldsymbol{x}(i) \cdot \boldsymbol{x}(i)^H = \beta \boldsymbol{C}_{xx}(n-1) + \boldsymbol{x}(i) \cdot \boldsymbol{x}(i)^H, \tag{3.6}$$

$$\boldsymbol{C}_{yx}(n) = \sum_{i=1}^{N} \beta^{n-i} \boldsymbol{y}(i) \cdot \boldsymbol{x}(i)^H = \beta \boldsymbol{C}_{yx}(n-1) + \boldsymbol{y}(i) \cdot \boldsymbol{x}(i)^H, \tag{3.7}$$

$$\boldsymbol{C}_{xy}(n) = \sum_{i=1}^{N} \beta^{n-i} \boldsymbol{x}(i) \cdot \boldsymbol{y}(i)^H = \beta \boldsymbol{C}_{xy}(n-1) + \boldsymbol{x}(i) \cdot \boldsymbol{y}(i)^H, \tag{3.8}$$

$$\boldsymbol{C}_{yy}(n) = \sum_{i=1}^{N} \beta^{n-i} \boldsymbol{y}(i) \cdot \boldsymbol{y}(i)^H = \beta \boldsymbol{C}_{yy}(n-1) + \boldsymbol{y}(i) \cdot \boldsymbol{y}(i)^H. \tag{3.9}$$

As shown in (3.5), the cost function $J(\boldsymbol{U}_s(n))$ signifies as a second order function of the elements $\boldsymbol{U}_s(n)$. Now, applying the derivation process on it yields a linear equation with a first order function of $\boldsymbol{U}_s(n)$ as

$$\nabla_{\boldsymbol{U}_s^*}(J(\boldsymbol{U}_s(n))) = \boldsymbol{U}_s(n) \cdot \boldsymbol{C}_{yy}(n) - \boldsymbol{C}_{xy}(n) = \boldsymbol{0}. \tag{3.10}$$

So, the cost function in (3.5) can be minimized via solving (3.10) and computing the inverse of the correlated matrix $\boldsymbol{C}_{yy}(n)$ according to

$$\boldsymbol{U}_s(n) = \boldsymbol{C}_{xy}(n) \cdot \boldsymbol{C}_{yy}^{-1}(n). \tag{3.11}$$

Considering to the mathematical complexity, we can see that the recursive computation of the correlation $\boldsymbol{U}_s(n)$ from $\boldsymbol{C}_{xy}(n) \in \mathbb{C}^{M \times d}$ and $\boldsymbol{C}_{yy}(n) \in \mathbb{C}^{d \times d}$ demands $O(Md^2) + O(d^3)$ operations in the complex domain. However, the complexity of calculating the inverse of $\boldsymbol{C}_{yy}(n)$ becomes in the $O(Md)$ operations by using the matrix

inverse lemma (Sherman-Morrison formula, see [PP12])

$$(\boldsymbol{A}+\boldsymbol{u}\boldsymbol{v}^H)^{-1}=\boldsymbol{A}^{-1}-\frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^H\boldsymbol{A}^{-1}}{1+\boldsymbol{v}^H\boldsymbol{A}^{-1}\boldsymbol{u}}. \tag{3.12}$$

By considering $\boldsymbol{P}(n-1)=\frac{1}{\beta}\boldsymbol{C}_{yy}^{-1}(n-1)$ and replacing it with $\boldsymbol{A}$ in (3.9), in addition to replace each of $\boldsymbol{u},$ and $\boldsymbol{v}^H$ with $\boldsymbol{y},$ and $\boldsymbol{y}^H$ in (3.9), respectively, we obtain

$$\boldsymbol{P}(n)=\frac{1}{\beta}(\boldsymbol{P}(n-1)-\frac{\boldsymbol{P}(n-1)\boldsymbol{y}(n)\boldsymbol{y}(n)^H\boldsymbol{P}(n-1)}{\beta+\boldsymbol{y}(n)^H\boldsymbol{P}(n-1)\boldsymbol{y}(n)}). \tag{3.13}$$

The PAST algorithm is summarized in Tab. 3.1, where Tri$\{\boldsymbol{P}\}$ denotes that the only upper (or lower) triangular part of $\boldsymbol{P}$ is computed and the Hermitian transposed version is copied to the other lower (or upper) triangular part, since $\boldsymbol{P}$ is a Hermitian matrix.

Note that the initial elements of the matrices $\boldsymbol{P}(0)$ and $\boldsymbol{U}_s(0)$ should be chosen suitably such that $\boldsymbol{P}(0)$ is a Hermitian matrix and $\boldsymbol{U}_s(0)$ is a unitary matrix.

As we can see from Tab. 3.1, the projection of the observation data vector is performed via the multiplication of the previous guessing of the signal subspace with the current data vector. This approximation in addition to use the RLS method decrease the mathematical complexity to $3Md+O(d^2)$ operations every update cycle.

Choose $\boldsymbol{P}(0)$ and $\boldsymbol{W}(0)$ suitably

FOR n = 1, 2, ... Do

$$\begin{aligned}
\boldsymbol{y}(n) &= \boldsymbol{U}_s(n-1)^{\mathrm{H}}\cdot\boldsymbol{x}(n)\\
\boldsymbol{h}(n) &= \boldsymbol{P}(n-1)\cdot\boldsymbol{y}(n)\\
\boldsymbol{g}(n) &= \boldsymbol{h}(n)/\big(\beta+\boldsymbol{y}(n)^{\mathrm{H}}\cdot\boldsymbol{h}(n)\big)\\
\boldsymbol{P}(n) &= \frac{1}{\beta}\cdot\mathrm{Tri}\{\boldsymbol{P}(n-1)-\boldsymbol{g}(n)\cdot\boldsymbol{h}(n)^{\mathrm{H}}\}\\
\boldsymbol{e}(n) &= \boldsymbol{x}(n)-\boldsymbol{U}_s(n-1)\cdot\boldsymbol{y}(n)\\
\boldsymbol{U}_s(n) &= \boldsymbol{U}_s(n-1)+\boldsymbol{e}(n)\cdot\boldsymbol{g}(n)^{\mathrm{H}}
\end{aligned}$$

Tab. 3.1: PAST algorithm for tracking the signal subspace

3.2 Extended PAST

The PAST algorithm has been extended by [CH13]. Moreover, a generic framework for tensor subspace tracking via Kronecker-structured projections (TeTraKron) has been performed depending on the extended version of the PAST algorithm.

For a data model of a multidimensional harmonic retrieval problem given according to (2.28), an observed tensor $\boldsymbol{\mathcal{X}}(n) \in \mathbb{C}^{M_1 \times M_2 \times \ldots M_R}$ at n-th snapshot can be expressed as

$$\boldsymbol{\mathcal{X}}(n) = \boldsymbol{\mathcal{A}}(n) \times_{R+1} \boldsymbol{s}^T(n) + \boldsymbol{\mathcal{W}}(n), \tag{3.14}$$

where $\boldsymbol{\mathcal{A}}(n) \in \mathbb{C}^{M_1 \times M_2 \times \ldots \times M_R \times d}$ is the array steering tensor at n-th snapshot, $\boldsymbol{s}(n) = [s_1(n), ..., s_d(n)]^T \in \mathbb{C}^d$ symbolizes a random sources vector with a non-singular correlation, $\boldsymbol{\mathcal{W}}(n)$ is an additive noise tensor with a same size like the observation data tensor at n-th snapshot as well.

By considering that the current r-mode unfoldings is given as $\tilde{\boldsymbol{X}}_r(n) = [\boldsymbol{\mathcal{X}}(n)]_{(r)} \in \mathbb{C}^{M_r \times \frac{M}{M_r}}, r = 1, 2, ...R$ and $M = \prod_{r=1}^{R} M_r$, the relation to the vector data concept at each snapshot can be obtain as

$$\text{vec}\{\tilde{\boldsymbol{X}}_r(n)\} = \boldsymbol{x}(n) = \text{vec}\{[\boldsymbol{\mathcal{X}}(n)]^T_{(R+1)}\}. \tag{3.15}$$

Tab. 3.2 summarizes the Extended PAST algorithm for tracking the r-mode unfolding subspaces, where the correlation matrices can be written in this way

$$\begin{aligned} \boldsymbol{C}_{xy_r}(n) &= \sum_{i=1}^{n} \beta^{n-i} \tilde{\boldsymbol{X}}_r(i) \boldsymbol{Y}_r^H(i) = \beta \boldsymbol{C}_{xy_r}(n-1) + \tilde{\boldsymbol{X}}_r(n) \boldsymbol{Y}_r^H(n), \\ \boldsymbol{C}_{yy_r}(n) &= \sum_{i=1}^{n} \beta^{n-i} \boldsymbol{Y}_r(i) \boldsymbol{Y}_r^H(i) = \beta \boldsymbol{C}_{yy_r}(n-1) + \boldsymbol{Y}_r(n) \boldsymbol{Y}_r^H(n). \end{aligned} \tag{3.16}$$

In order to calculate the inverse of matrix $\boldsymbol{C}_{yy_r}(n)$ each update, [CH13] used the matrix inversion lemma (Woodbury matrix identity, see [GL96]) lowering in this step the complexity. The Woodbury identity comes in many variants. The latter of the two can be found in [GL96] as

$$\begin{aligned} (\boldsymbol{A} + \boldsymbol{U}\boldsymbol{B}\boldsymbol{U}^H)^{-1} &= \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{B}^{-1} + \boldsymbol{U}^H\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{U}^H\boldsymbol{A}^{-1}, \\ (\boldsymbol{A} + \boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^H)^{-1} &= \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{B}^{-1} + \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}\boldsymbol{A}^{-1}. \end{aligned} \tag{3.17}$$

Now, by replacing $\boldsymbol{A}$, $\boldsymbol{U}$, and $\boldsymbol{B}$ in (3.17) with $\boldsymbol{P}_r(n-1)$, $\boldsymbol{Y}_r$, and $\boldsymbol{I}_{M/M_r}$, respectively,

we obtain the recursive computation of $\boldsymbol{P}_r(n)$ as

$$\begin{aligned}\boldsymbol{P}_r(n) &= \frac{1}{\beta}\boldsymbol{P}_r(n-1) - \frac{1}{\beta}\boldsymbol{P}_r(n-1)\boldsymbol{Y}_r(n) \\ &\cdot \quad (\beta I_{M/M_r} + \boldsymbol{Y}_r(n)^H \boldsymbol{P}_r(n-1)\boldsymbol{Y}_r(n))^{-1}\boldsymbol{Y}_r(n)^H \boldsymbol{P}_r(n-1), \end{aligned} \tag{3.18}$$

with $r = 1, 2, ..., R$.

$\boldsymbol{P}_r(0) = \boldsymbol{I}_{d\times d}$, $\boldsymbol{U}_r^{[s]}(0) = \boldsymbol{I}_{M\times d}$

FOR n = 1, 2, ... DO

$$\begin{aligned}\boldsymbol{Y}_r(n) &= \boldsymbol{U}_r^{[s]}(n-1)^{\mathrm{H}} \cdot \tilde{\boldsymbol{X}}_r(n) \\ \boldsymbol{H}_r(n) &= \boldsymbol{P}_r(n-1) \cdot \boldsymbol{Y}_r(n) \\ \boldsymbol{G}_r(n) &= \boldsymbol{H}_r(n)/\big(\beta \cdot \boldsymbol{I}_{M/M_r} + \boldsymbol{Y}_r(n)^{\mathrm{H}} \cdot \boldsymbol{H}_r(n)\big) \\ \boldsymbol{P}_r(n) &= \frac{1}{\beta} \cdot \mathrm{Tri}\{\boldsymbol{P}_r(n-1) - \boldsymbol{G}_r(n) \cdot \boldsymbol{H}_r(n)^{\mathrm{H}}\} \\ \boldsymbol{E}_r(n) &= \tilde{\boldsymbol{X}}_r(n) - \boldsymbol{U}_r^{[s]}(n-1) \cdot \boldsymbol{Y}_r(n) \\ \boldsymbol{U}_r^{[s]}(n) &= \boldsymbol{U}_r^{[s]}(n-1) + \boldsymbol{E}_r(n) \cdot \boldsymbol{G}_r(n)^{\mathrm{H}}\end{aligned}$$

Tab. 3.2: Extended PAST algorithm

After tracking the r-mode unfolding subspaces $\boldsymbol{U}_r^{[s]}(n)$ as well as the subspace $\boldsymbol{U}_s$ of the data vector $\boldsymbol{x}(n)$ according to the Extended PAST and PAST algorithms, [CH13] used the Kronecker structured projection to extract the tensor-based subspaces as

$$\left[\hat{\boldsymbol{\mathcal{U}}}^{[s]}\right]^T_{(R+1)} = \left(\boldsymbol{U}_1^{[s]} \otimes \boldsymbol{U}_2^{[s]} ... \otimes \boldsymbol{U}_R^{[s]}\right) \cdot \bar{\boldsymbol{U}}_s, \tag{3.19}$$

where, $\bar{\boldsymbol{U}}_s = \left(\boldsymbol{U}_1^{[s]} \otimes \boldsymbol{U}_2^{[s]} ... \otimes \boldsymbol{U}_R^{[s]}\right)^H \cdot \hat{\boldsymbol{U}}_s \in \mathbb{C}^{d^R \times d}$.

Tensor-based signal subspace tracking via Kronecker structured projections (TeTraKron) increased the mathematical complexity to $3Md + \max\left(O(\frac{M}{M_r})^3\right)$ at each update. However, TeTraKron based tracking schemes outperformed the PAST algorithm in time-varying scenarios as shown in [CH13].

In the next step, we are going to incorporate the FBA processing onto the PAST algorithm and extend it again for the tensor case, respectively.

3.3 FBA-PAST Algorithm

Let $\boldsymbol{X} \in \mathbb{C}^{M\times N}$ be an observation data over an N window length according to (2.24). The extended matrix $\boldsymbol{Z} \in \mathbb{C}^{M\times 2N}$ can be performed according to [HN95] as

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{\Pi}_M \boldsymbol{X}^* \boldsymbol{\Pi}_N. \end{bmatrix} \tag{3.20}$$

Since $\boldsymbol{Z}$ is now a centro-Hermitian matrix, the real-valued processing can be applied over an N window length as follows

$$\varphi(\boldsymbol{Z}) = \boldsymbol{Q}_M^H \cdot \boldsymbol{Z} \cdot \boldsymbol{Q}_{2N} \in \mathbb{R}^{M\times 2N}, \tag{3.21}$$

where $\boldsymbol{Q}_p \in \mathbb{C}^{p\times p}$ is a left $\boldsymbol{\Pi}$-real unitary matrix. In our study, we selected $\boldsymbol{Q}_p \in \mathbb{C}^{p\times p}$ to be a left $\boldsymbol{\Pi}$-real unitary matrix according to [HN95]. Fig. 3.2 visualizes the FBA version in a block diagram.

According to [HN95], the relation between the columns subspace $\boldsymbol{U}_s \in \mathbb{C}^{M\times d}$ of the data matrix $\boldsymbol{X}$ and the d dominant left singular vectors $\boldsymbol{E}_s \in \mathbb{R}^{M\times d}$ of the extended data matrix $\boldsymbol{Z}$ is calculated as

$$\boldsymbol{U}_s = \boldsymbol{Q}_M \boldsymbol{E}_s. \tag{3.22}$$

Now, to obtain the FBA-PAST algorithm , let us assume that the extended data

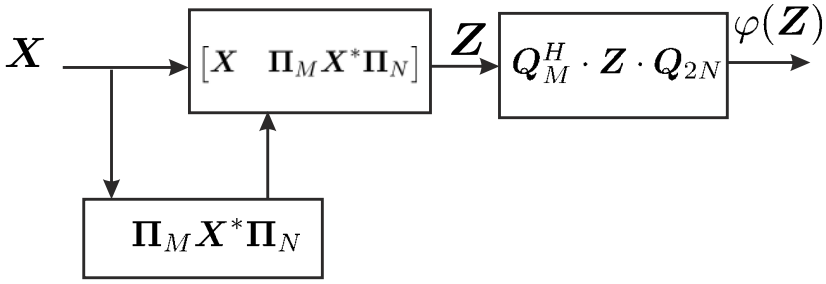


Fig. 3.2: FBA block diagram. $\boldsymbol{X} \in \mathbb{C}^{M\times N}$ represents the observation data, $\boldsymbol{Z} = [\boldsymbol{X} \;\; \boldsymbol{\Pi}_M \boldsymbol{X}^* \boldsymbol{\Pi}_N] \in \mathbb{C}^{M\times 2N}$ is the centro-symmetric matrix, and $\varphi(\boldsymbol{Z}) = \boldsymbol{Q}_M^H \cdot \boldsymbol{Z} \cdot \boldsymbol{Q}_{2N} \in \mathbb{R}^{M\times 2N}$ symbolizes the mapping matrix.

$\tilde{\boldsymbol{Z}}(n) \in \mathbb{C}^{M\times 2}$ at n-th snapshots can be projected into a matrix $\tilde{\boldsymbol{Y}}(n) \in \mathbb{R}^{M\times 2}$ as

$$\tilde{\boldsymbol{Y}}(n) \;\; = \;\; \boldsymbol{E}_s(n-1)^{\mathrm{H}} \cdot \varphi(\tilde{\boldsymbol{Z}}(n)). \tag{3.23}$$

Then, the FPA-PAST algorithm can be summarized in a few steps as shown in

Tab. 3.3, where the correlation matrices in FAB-PAST algorithm are computed as

$$\begin{aligned}
\boldsymbol{C}_{zz}(n) &= \sum_{i=1}^{N} \beta^{n-i} \varphi(\tilde{\boldsymbol{Z}}(i)) \cdot \varphi(\tilde{\boldsymbol{Z}}(i))^H = \beta \boldsymbol{C}_{zz}(n-1) + \varphi(\tilde{\boldsymbol{Z}}(n)) \cdot \varphi(\tilde{\boldsymbol{Z}}(n))^H, \\
\boldsymbol{C}_{yz}(n) &= \sum_{i=1}^{N} \beta^{n-i} \tilde{\boldsymbol{Y}}(i) \cdot \varphi(\tilde{\boldsymbol{Z}}(i))^H = \beta \boldsymbol{C}_{yz}(n-1) + \tilde{\boldsymbol{Y}}(n) \cdot \varphi(\tilde{\boldsymbol{Z}}(n))^H, \\
\boldsymbol{C}_{zy}(n) &= \sum_{i=1}^{N} \beta^{n-i} \varphi(\tilde{\boldsymbol{Z}}(i)) \cdot \tilde{\boldsymbol{Y}}(i)^H = \beta \boldsymbol{C}_{zy}(n-1) + \varphi(\tilde{\boldsymbol{Z}}(n)) \cdot \tilde{\boldsymbol{Y}}^H(n), \\
\boldsymbol{C}_{yy}(n) &= \sum_{i=1}^{N} \beta^{n-i} \tilde{\boldsymbol{Y}}(i) \cdot \tilde{\boldsymbol{Y}}(i)^H = \beta \boldsymbol{C}_{yy}(n-1) + \tilde{\boldsymbol{Y}}(n) \cdot \tilde{\boldsymbol{Y}}^H(n). \qquad (3.24)
\end{aligned}$$

Similar to the previous section, we use the matrix inversion lemma (Woodbury matrix identity) to calculate the inverse of matrix $\boldsymbol{C}_{yy}(n)$ at each update to lower the complexity. Hence, replacing $\boldsymbol{A}$, $\boldsymbol{U}$, and $\boldsymbol{B}$ in (3.17) with $\boldsymbol{P}^{\mathrm{FBA}}(n-1)$, $\tilde{\boldsymbol{Y}}(n)$, and $\boldsymbol{I}_2$, respectively, yields the recursive computation of $\boldsymbol{P}^{\mathrm{FBA}}(n)$ as

$$\begin{aligned}
\boldsymbol{P}^{\mathrm{FBA}}(n) &= \frac{1}{\beta}\boldsymbol{P}^{\mathrm{FBA}}(n-1) - \frac{1}{\beta}\boldsymbol{P}^{\mathrm{FBA}}(n-1)\tilde{\boldsymbol{Y}}(n) \\
&\cdot \quad (\beta I_2 + \tilde{\boldsymbol{Y}}(n)^H \boldsymbol{P}^{\mathrm{FBA}}(n-1)\tilde{\boldsymbol{Y}}(n))^{-1} \tilde{\boldsymbol{Y}}(n)^H \boldsymbol{P}^{\mathrm{FBA}}(n-1), \qquad (3.25)
\end{aligned}$$

Choose $\boldsymbol{P}^{\mathrm{FBA}}(0)$ and $\boldsymbol{E}_s(0)$ suitably

FOR n = 1, 2, ... Do

$$\begin{aligned}
\tilde{\boldsymbol{Y}}(n) &= \boldsymbol{E}_s(n-1)^{\mathrm{H}} \cdot \varphi(\tilde{\boldsymbol{Z}}(n)) \in \mathbb{R}^{d\times 2} \\
\boldsymbol{H}(n) &= \boldsymbol{P}(n-1) \cdot \tilde{\boldsymbol{Y}}(n) \in \mathbb{R}^{d\times 2} \\
\boldsymbol{G}(n) &= \boldsymbol{H}(n) \cdot \Big(\beta \boldsymbol{I}_2 + \tilde{\boldsymbol{Y}}(n)^{\mathrm{H}} \cdot \boldsymbol{H}(n)\Big)^{-1} \in \mathbb{R}^{d\times 2} \\
\boldsymbol{P}^{\mathrm{FBA}}(n) &= \frac{1}{\beta} \cdot \mathrm{Tri}\{\boldsymbol{P}^{\mathrm{FBA}}(n-1) - \boldsymbol{G}(n) \cdot \boldsymbol{H}(n)^{\mathrm{H}}\} \in \mathbb{R}^{d\times d} \\
\boldsymbol{B}(n) &= \varphi(\tilde{\boldsymbol{Z}}(i)) - \boldsymbol{E}_s(n-1) \cdot \tilde{\boldsymbol{Y}}(n) \in \mathbb{R}^{M\times 2} \\
\boldsymbol{E}_s(n) &= \boldsymbol{E}_s(n-1) + \boldsymbol{B}(n) \cdot \boldsymbol{G}(n)^{\mathrm{H}} \in \mathbb{R}^{M\times d}
\end{aligned}$$

Tab. 3.3: FBA-PAST algorithm

The functions as shown in Tab. 3.3 have been changed from the vectors of one dimension to matrices of double sizes in the real domain.

The initial values have been chosen such as $\boldsymbol{P}(0)$ is a Hermitian positive definite matrix, and $\boldsymbol{E}_s(0)$ contains d orthonormal vectors, i.e., $\boldsymbol{P}(0) = I_d$ and $\boldsymbol{E}_s(0) = I_{M\times d}$. We can see that duplicating the observation data by incorporating the FBA processing, the FBA-PAST algorithm requires $3Md + O(d^2)$ operators each update involving only real-valued computations.

In the next, we are going to extend the FPA-PAST algorithm for the data model based tensor approach to track the real-valued n-mode unfolding subspaces.

3.4 Extended FBA-PAST Algorithm

Similar to the Extended PAST algorithm, we consider a data model of a multidimensional harmonic retrieval problem according to (2.28). Then, the extended tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{M_1\times M_2\times\ldots\times M_R\times 2N}$ can be given according to [HRDG08] as

$$\boldsymbol{\mathcal{Z}} = \left[\boldsymbol{\mathcal{X}} \sqcup_{R+1} \left(\boldsymbol{\mathcal{X}}^{*} \times_1 \boldsymbol{\Pi}_{M_1} \times_2 \boldsymbol{\Pi}_{M_2} \ldots \times_{R+1} \boldsymbol{\Pi}_N\right)\right]. \tag{3.26}$$

Consequently, the centro-Hermitian tensor $\boldsymbol{\mathcal{Z}}$ is mapped onto the set of real-valued tensor using the following transformation

$$\varphi(\boldsymbol{\mathcal{Z}}) = \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{Q}_{M_1}^{H} \times_2 \boldsymbol{Q}_{M_2}^{H} \ldots \times_{R+1} \boldsymbol{Q}_{2N}^{H} \in \mathbb{R}^{M_1\times M_2\times\ldots M_R\times 2N}, \tag{3.27}$$

where the $\boldsymbol{Q}_p \in \mathbb{C}^{p\times p}$ is a left $\boldsymbol{\Pi}$-real unitary matrix.

According to [HRDG08], applying the HOSVD procedure on $\varphi(\boldsymbol{\mathcal{Z}})$ yields

$$\begin{aligned}
\varphi(\boldsymbol{\mathcal{Z}}) &= \boldsymbol{\mathcal{S}}_Z \times_1 \boldsymbol{E}_1 \times_2 \boldsymbol{E}_2 \ldots \times_R \boldsymbol{E}_R \times_{R+1} \boldsymbol{E}_{R+1}, \\
&\approx \boldsymbol{\mathcal{S}}_Z^{[s]} \times_1 \boldsymbol{E}_1^{[s]} \times_2 \boldsymbol{E}_2^{[s]} \ldots \times_R \boldsymbol{E}_R^{[s]} \times_{R+1} \boldsymbol{E}_{R+1}^{[s]} \\
&\approx \boldsymbol{\mathcal{E}}^{[s]} \times_{R+1} \boldsymbol{E}_{R+1}^{[s]},
\end{aligned} \tag{3.28}$$

where $\boldsymbol{\mathcal{E}}^{[s]} = \boldsymbol{\mathcal{S}}_Z^{[s]} \times_1 \boldsymbol{E}_1^{[s]} \times_2 \boldsymbol{E}_2^{[s]} \ldots \times_R \boldsymbol{E}_R^{[s]}$ represents the multidimensional extension of the real-valued basis $\boldsymbol{E}_s$ known from the matrix approach.

Again, by considering that the current r-mode unfoldings of the extended tensor can be given as $\boldsymbol{\Xi}(n)_r = [\varphi(\boldsymbol{\mathcal{Z}}(n)]_{(r)} \in \mathbb{C}^{M_r\times 2\frac{M}{M_r}}, r = 1, 2, .., R$ and $M = \prod_{r=1}^{R} M_r$, we can modify the Extended PAST algorithm to the Extended FBA-PAST algorithm where the projection approximation matrices are denoted as $\boldsymbol{L}(n)_r = \boldsymbol{E}_r(n-1)^H \cdot \boldsymbol{\Xi}(n)_r \in \mathbb{R}^{d\times 2\frac{M}{M_r}}$.

Tab. 3.4 illustrates the Extended FBA-PAST algorithm for tracking the real-valued

r-mode unfoldings, where the correlation matrices can be computed in this way

$$\begin{aligned} \boldsymbol{C}_{\xi l_r}(n) &= \sum_{i=1}^{n} \beta^{n-i} \boldsymbol{\Xi}_r(i) \boldsymbol{L}_r(i)^H = \beta \boldsymbol{C}_{\xi l_r}(n-1) + \boldsymbol{\Xi}_r(n) \boldsymbol{L}_r(n)^H, \\ \boldsymbol{C}_{l l_r}(n) &= \sum_{i=1}^{n} \beta^{n-i} \boldsymbol{L}_r(i) \boldsymbol{L}_r(i)^H = \beta \boldsymbol{C}_{l l_r}(n-1) + \boldsymbol{L}_r(n) \boldsymbol{L}_r(n)^H. \end{aligned} \tag{3.29}$$

In order to calculate the inverse of $\boldsymbol{C}_{ll_r}(n)$ each update, we use also the matrix inversion lemma (Woodbury matrix identity) in (3.17). Thereby, replacing $\boldsymbol{A}$, $\boldsymbol{U}$, and $\boldsymbol{B}$ in (3.17) with $\boldsymbol{P}_r^{\mathrm{FPA}}(n-1)$, $\boldsymbol{Y}_r$, and $\boldsymbol{I}_{M/M_r}$, respectively, give the recursive computation of $\boldsymbol{P}_r^{\mathrm{FPA}}(n)$ as

$$\begin{aligned} \boldsymbol{P}_r^{\mathrm{FPA}}(n) &= \frac{1}{\beta} \boldsymbol{P}_r^{\mathrm{FPA}}(n-1) - \frac{1}{\beta} \boldsymbol{P}_r^{\mathrm{FPA}}(n-1) \boldsymbol{L}_r(n) \\ &\cdot \quad (\beta I_{2M/M_r} + \boldsymbol{L}_r(n)^H \boldsymbol{P}_r^{\mathrm{FPA}}(n-1) \boldsymbol{L}_r(n))^{-1} \\ &\cdot \quad \boldsymbol{L}_r(n)^H \boldsymbol{P}_r^{\mathrm{FPA}}(n-1), \end{aligned} \tag{3.30}$$

with $r = 1, 2, ..., R$.

Choose $\boldsymbol{P}_r^{\mathrm{FBA}}(0)$ and $\boldsymbol{E}_r(0)$ suitably

FOR n = 1, 2, ... Do

$$\begin{aligned} \boldsymbol{L}_r(n) &= \boldsymbol{E}_r(n-1)^{\mathrm{H}} \cdot \boldsymbol{\Xi}_r(n) \in \mathbb{R}^{d \times 2\frac{M}{M_r}} \\ \boldsymbol{H}_r(n) &= \boldsymbol{P}_r(n-1) \cdot \boldsymbol{L}_r(n) \in \mathbb{R}^{d \times 2\frac{M}{M_r}} \\ \boldsymbol{G}_r(n) &= \boldsymbol{H}_r(n) / \big(\beta \cdot \boldsymbol{I}_{2M/M_r} + \boldsymbol{L}_r(n)^{\mathrm{H}} \cdot \boldsymbol{H}_r(n)\big) \in \mathbb{R}^{d \times 2\frac{M}{M_r}} \\ \boldsymbol{P}_r^{\mathrm{FBA}}(n) &= \frac{1}{\beta} \cdot \mathrm{Tri}\{\boldsymbol{P}_r^{\mathrm{FBA}}(n-1) - \boldsymbol{G}_r(n) \cdot \boldsymbol{H}_r^{\mathrm{H}}(n)\} \in \mathbb{R}^{d \times d} \\ \boldsymbol{B}_r(n) &= \boldsymbol{\Xi}_r(n) - \boldsymbol{E}_r(n-1) \cdot \boldsymbol{L}_r(n) \in \mathbb{R}^{M_r \times 2\frac{M}{M_r}} \\ \boldsymbol{E}_r(n) &= \boldsymbol{E}_r(n-1) + \boldsymbol{B}_r(n) \cdot \boldsymbol{G}_r(n)^{\mathrm{H}} \in \mathbb{R}^{M_r \times d} \end{aligned}$$

Tab. 3.4: Extended FBA-PAST algorithm

After tracking the real-valued r-mode unfolding subspaces $\boldsymbol{E}_r(n)$ as well as the real-valued subspace $\boldsymbol{E}_s$ of the mapping matrix $\varphi(\boldsymbol{Z})$, we use the TeTraKron algorithm to obtain the tensor-based subspace tracking each update as follows

$$\left[\hat{\boldsymbol{\mathcal{E}}}^{[s]}\right]^{T}_{(R+1)} = \underbrace{\boldsymbol{Q}_{Kron} \cdot \left(\hat{\boldsymbol{E}}_1^{[s]} \otimes \hat{\boldsymbol{E}}_2^{[s]} ... \otimes \hat{\boldsymbol{E}}_R^{[s]}\right)}_{\tilde{\boldsymbol{E}}_2^{[s]}} \cdot \underbrace{\left(\hat{\boldsymbol{E}}_1^{[s]} \otimes \hat{\boldsymbol{E}}_2^{[s]} ... \otimes \hat{\boldsymbol{E}}_R^{[s]}\right)^{H} \cdot \boldsymbol{Q}_{Kron}^{H}}_{\tilde{\boldsymbol{E}}_2^{[s]H}} \hat{\boldsymbol{E}}_s \quad (3.31)$$

$$= \tilde{\boldsymbol{E}}_{Kron}^{[s]} \cdot \underbrace{\tilde{\boldsymbol{E}}_{Kron}^{[s]H} \hat{\boldsymbol{E}}_s}_{\hat{\tilde{\boldsymbol{E}}}_s} \quad (3.32)$$

$$= \tilde{\boldsymbol{E}}_{Kron}^{[s]} \cdot \hat{\tilde{\boldsymbol{E}}}_s. \quad (3.33)$$

According to (3.33), the Extended FBA-PAST algorithm with TeTraKron algorithm has the same mathematical operations like the Extended PAST algorithm with TeTraKron. However, all the computations are performed in the real domain, so the complexity is less than the similar algorithm based on the Extended PAST algorithm.

3.5 Simulation Results

In the following, we show some simulation results about the performance of the FBA-PAST and the Extended FBA-PAST algorithms compared to the PAST and Extended PAST algorithms. Depending on the movement of the sources, we selected different cases related to the correlation between the sources and the SNR.

In order to evaluate the performance of our algorithms, we used the Large Principal Angle (LPA) and the Root Mean Square Error (RMSE) concepts. LPA measures the distance between two subspaces according to [GL96] as

$$\text{LPA} = \cos^{-1}(\sigma_{\min}\left\{\text{orth}\left\{\boldsymbol{U}_1\right\}^{H} \cdot \text{orth}\left\{\boldsymbol{U}_2\right\}\right\}), \quad (3.34)$$

where orth $\{\boldsymbol{U}_i\}$ is an orthonormal basis, and $\sigma_{\min}\{\boldsymbol{T}\}$ gives the smallest singular value of the matrix $\boldsymbol{T}$. According to [HRDG08], the second criterion is the total RMSE that measures the error between the original value and the estimated one as

$$\text{RMSE}_{\text{tot}} = \sqrt{\text{E}\left\{\sum_{r=1}^{R}\sum_{i=1}^{d}(\mu_i^{(r)} - \hat{\mu}_i^{(r)})^2\right\}}, \quad (3.35)$$

where $\hat{\mu}_i^{(r)}$ denotes an estimate of $\mu_i^{(r)}$.

Hence, calculating the RMSE requires tracking the DOA at each snapshot, we used the Standard Tensor ESPRIT (STE) with LS as well as Unitary Tensor ESPRIT (UTE) with LS according to [HRDG08] based on Extended PAST and Extended FBA-PAST

with TeTraKron, respectively. And we compared the performance of the aforementioned algorithms to 2-D ESPRIT and 2-D Unitary ESPRIT based on the PAST and BA-PAST algorithms in [SK93] and [HZMN95], respectively.

The scenarios are considered for $d = 3$ sources in the far field criteria with narrowband signals impinging on a URA of 6×6 sensors as visualized in Fig.3.3. The distance between the adjacent sensors has a half of the wavelength $\lambda/2$ in the vertical and horizontal dimensions, respectively.

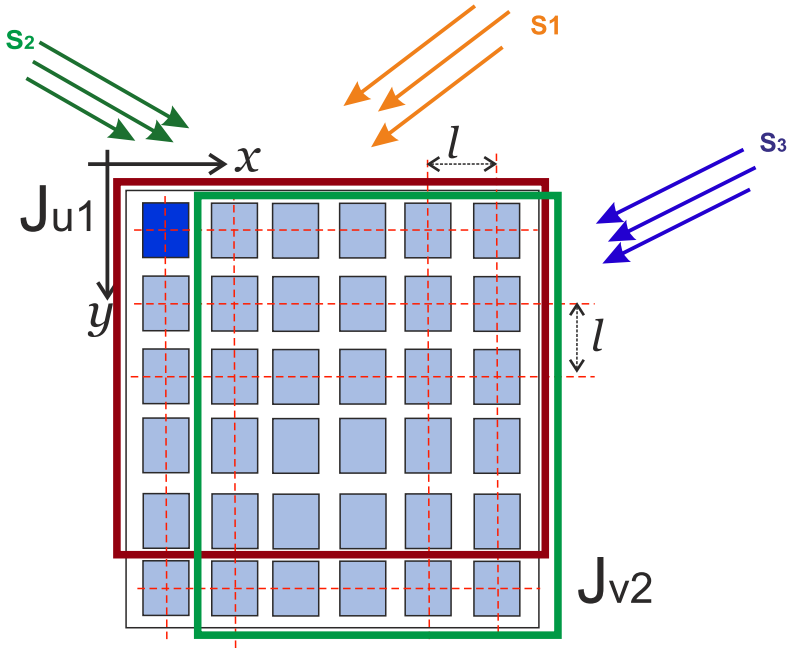


Fig. 3.3: URA of $M = 6 \times 6$ identical sensors with maximum overlapping selection matrices $\boldsymbol{J}_{u_1}$, $\boldsymbol{J}_{v_1}$ in the vertical and horizontal planes, respectively

In addition, the forgetting factor β has been chosen to be 0.97 at length window $N = 1000$, according to [Yan95]. The sources and the noise samples are considered to be a zero mean circularly symmetric complex Gaussian distribution. The sequence of snapshots $\boldsymbol{\mathcal{X}}(i), i = 1, 2, ...$ is generated according to the data model (2.28).

The initial values $\boldsymbol{P}(0)$, $\boldsymbol{U}_s(0)$, $\boldsymbol{P}_r(0)$, $\boldsymbol{U}_r^{[s]}(0)$, $\boldsymbol{P}(0)^{FBA}$, $\boldsymbol{E}_s(0)$, $\boldsymbol{P}_r^{FBA}(0)$ and $\boldsymbol{E}_r^{[s]}(0)$ have to be chosen such as, the projection matrices must be a Hermitian positive definite, e.g., $d \times d$ identity matrix; and the signal subspaces in addition to the real-valued signal subspaces contain d orthonormal vectors.

We should note that, the initial values impact the transient behavior of the algorithm. However, the performance of the algorithm remains stable.

The DOA can be computed as $\mu_i = \pi \cos\phi_i \sin\theta_i$ and $\nu_i = \pi \sin\phi_i \sin\theta_i$. Then $\mu_i = 2\pi u_i$ and $\nu_i = 2\pi v_i$ with $|u_i| \leq 1/2$ and $|v_i| \leq 1/2$. In our study, we are going

to track the spatial frequencies μ_i and ν_i instead of tracking ϕ_i and θ_i, respectively.

Scenario 1

In this scenario, we consider two of sources to be slowly moving overall snapshots and the third one has a constant DOA, as shown in Fig. 3.4. The spatial frequencies μ_i and ν_i are given in the radian unit as

$$\begin{aligned} \mu_1[n] &= 2\pi(0.3 - 0.1 \cdot t[n]), \quad \mu_2[n] = 2\pi(0.2 - 0.1 \cdot t[n]), \quad \mu_3[n] = 2\pi(0.1), \\ \nu_1[n] &= 2\pi(0.3 - 0.1 \cdot t[n]), \quad \nu_2[n] = 2\pi(0.3 - 0.1 \cdot t[n]), \quad \nu_2[n] = 2\pi(0.1), \end{aligned} \tag{3.36}$$

where $n = 1, 2, ..., N$.

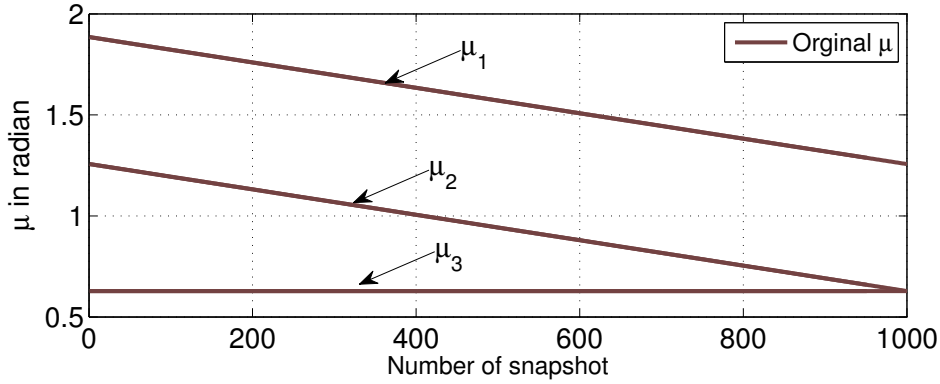
Now, we study the performance of our algorithms for three cases related to the SNR and the correlation between the sources.

The first case show the LPA and the RMSE verses the number of snapshots for $\rho = 0$ and SNR $= 0$ dB as depicted in Fig. 3.5 and Fig. 3.6, respectively. In Fig. 3.5, the curves are labeled as PAST (blue solid line), Extended PAST (cyan dashed line), FBA-PAST (red solid line), and Extended FBA-PAST (magenta dotted line) related to name of the aforementioned subspaces tracking algorithms, respectively.

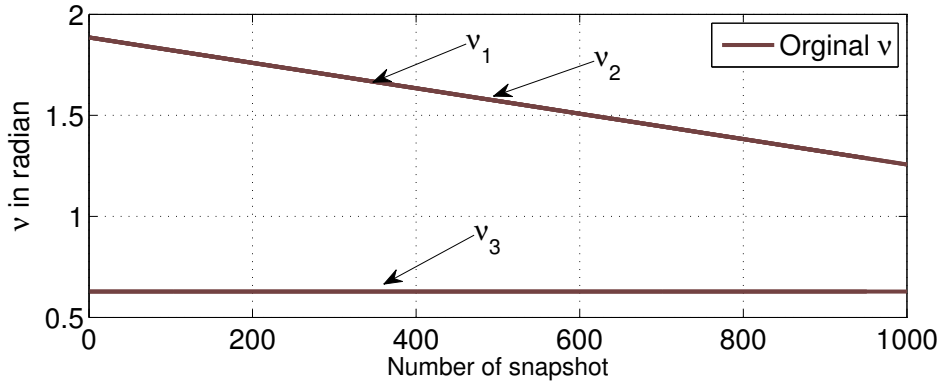
Moreover, the estimated spatial frequencies have been depicted in Fig. 3.6. Here, the curves are labeled as 2-D ESPRI+PAST (blue solid line), STE+Extended PAST (cyan dashed line), 2-D Unitary ESPRI+FBA-PAST (red solid line), and UTE+Extended FBA-PAST (magenta dotted line) related to name of the aforementioned DOA's tracking algorithms, respectively. It can be observed that the FBA-PAST algorithm improves the tracking of the signal subspace as well as the DOA's tracking. Hence, applying the real-valued processing duplicates the number of observations at each snapshot, the estimated signal subspaces for a few number of observation signals outperforms its counterpart without using FBA processing.

It can be seen also that the all algorithms are slightly different after a few number of observation signals tracking precisely the DOA for all the 3 sources. In the second case, we increased the correlation between the all sources such as $\rho = 0.99$ at the same SNR, as shown in Fig. 3.7 and Fig. 3.8, respectively. In fact, this kind of scenario can be considered as one of critical scenarios related to the high correlations between the all sources. It can be observed that the PAST algorithm completely failed in tracking the signal subspace. More precisely, it failed in tracking two of the sources as illustrated in Fig. 3.8.

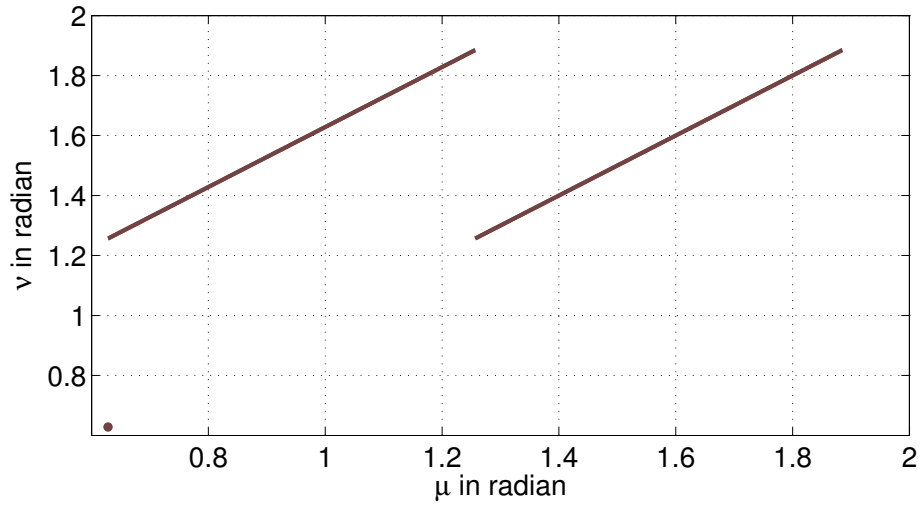
It can be found also that using the tenor-based subspaces improved the DOA tracking a bit. However the FBA-PAST algorithm gives better tracking than the Extended



(a) Original spatial frequencies on the horizontal plane



(b) Original spatial frequencies on the vertical plane



(c) ν vs. μ

Fig. 3.4: Spatial frequencies of 3 sources vs. the number of snapshots for scenario 1

PAST as well as the PAST algorithm. Moreover, the FBA-PAST algorithm required long period until it was able to track correctly the DOA.

Now coming to our contribution by incorporating the FBA to the Extended PAST algorithm. It can be seen obviously that the UTE based on Extended FBA-PAST with

LS outperforms the all algorithm above in the case of high correlation between the sources. Deep insight can be shown in Fig. 3.8 which depicts that the Extended FBA-PAST algorithm is able to track the all three sources after a few number of snapshots in spite of existing a high correlation between the sources.

Last, we evaluated the case of low SNR and no correlation between the sources signals. It can be observed from Fig. 3.10 that the Extended FBA-PAST outperforms the all algorithms above due to the denoising structure and using the FBA processor as well. The interesting results can be obtain from Fig. 3.9 and Fig. 3.10 such as, the all algorithms are more robustness than decreasing the signal energy, i.e., all algorithms are succeed in tracking the DOA for all sources at SNR = -3 dB after a few number of observations.

Scenario 2 Similarly to [CH13] scenario, we are going to track the DOA of 3 sources, two of them are crossing at the half of distance and the third has a constant DOA, as following

$$\begin{aligned} \mu_1[n] &= 2\pi(0.3 - 0.1 \cdot t[n]), \quad \mu_2[n] = 2\pi(0.2 + 0.1 \cdot t[n]), \quad \mu_3[n] = 2\pi(0.1), \\ \nu_1[n] &= 2\pi(0.3 - 0.1 \cdot t[n]), \quad \nu_2[n] = 2\pi(0.3 - 0.1 \cdot t[n]), \quad \nu_2[n] = 2\pi(0.1). \end{aligned} \tag{3.37}$$

Fig. 3.11 visualizes the spatial frequencies of the sources in the horizontal and vertical planes, respectively.

In this part, we study again more critical three cases related to the correlation between the sources and various SNR in addition to the correlation between the spatial frequencies. For the first case that has no correlation between the sources at SNR = 0 dB, Fig. 3.12 and Fig. 3.13 depict the simulation results, respectively. Similar to Scenario 1. Fig. 3.12 shows the LPA and RMSE verses the number of snapshots, respectively. And the estimated spatial frequencies in the horizontal and vertical planes are visualized as well in Fig. 3.13 (a) and (b), respectively. It can be seen that all algorithms can precisely track the signal subspaces as well as the DOA's of the three sources after a few number of observations. However, at the critical position, i.e., the crossing point between DOA of two sources or high correlation between the spatial frequencies of two sources, the PAST algorithm failed completely in tracking the signal subspaces as well as the DOA. And the FBA-PAST algorithm gave better results compare to PAST algorithm but also not enough to track any source at the critical position. Moreover, the UTE based on Extended FBA-PAST was able to tracke the DOA of all sources while STE based on the Extended PAST algorithm was able to track two of sources perfectly and was close from tracking the third one. This related to the denoising structure by using tenor-based subspaces tracking, and using the FBA

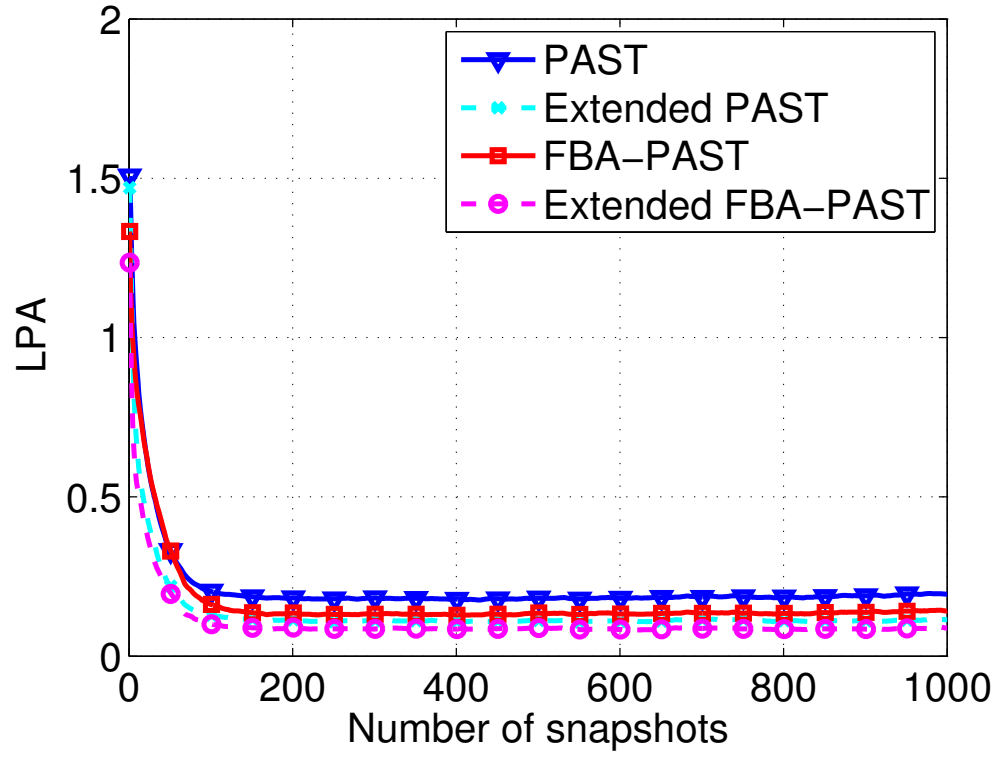
processing that involving real-valued computations and increasing the accuracy.

Consequently, we increase the correlation between the all sources to be $\rho = 0.5$ at the same SNR. As shown in Fig. 3.14 and Fig. 3.15, respectively, the PAST algorithm failed completely in tracking the signal subspaces. More precisely, It failed in tracking all the sources at the critical position and it took long time until it was able to resume the tracking when the sources were spatially separated again, as illustrated in Fig. 3.15(a) and (b), respectively. It can be observed also that using the tenor-based subspace tracking improves the DOA tracking a bit. In addition, the FBA-PAST algorithm gives again better performance than the Extended PAST as well as the PAST algorithm. Thus related to duplicate the observations each update by incorporating the FBA processing involving also only real-valued computations.

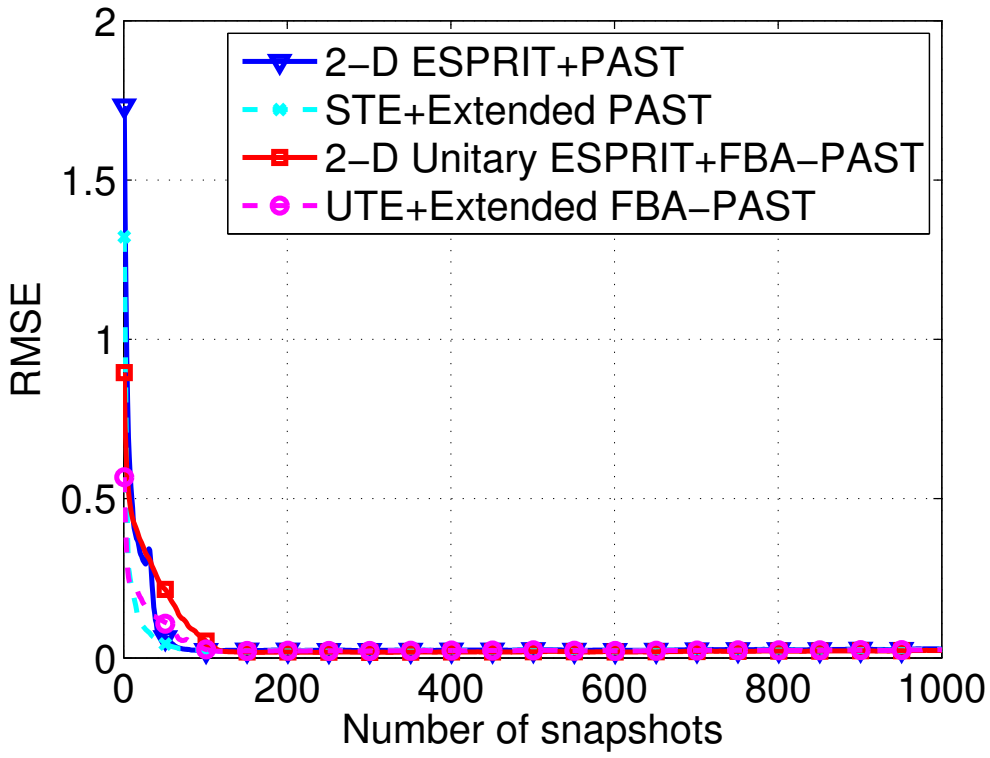
Now coming to our contribution by using the Extended FBA-PAST algorithm, it can be seen that our algorithm outperforms the all algorithms above in the case of high correlation between the sources. Deep insight can be shown in Fig. 3.15(a) and (b), respectively. Here, the Extended FBA-PAST algorithm was able to track the two sources correctly after a few number of snapshots and it was close from tracking the third source at the critical position. Thus related to use the denosing structure as well as to duplicate the observations by incorporating the FBA processing involving also only real-valued computations.

Last, we evaluate the performance of our algorithms at SNR $= -3$ dB with no correlation between the source signals. It can be found that the Extended FBA-PAST outperforms the all algorithms above at low SNR as shown in Fig. 3.16 as well as in Fig. 3.17.

In the next chapter, we are going to adaptive a new version of ESPRIT based on incorporating the FBA processing to it, then extend an adaptive Unitary ESPRIT from 1-D to 2-D, respectively.

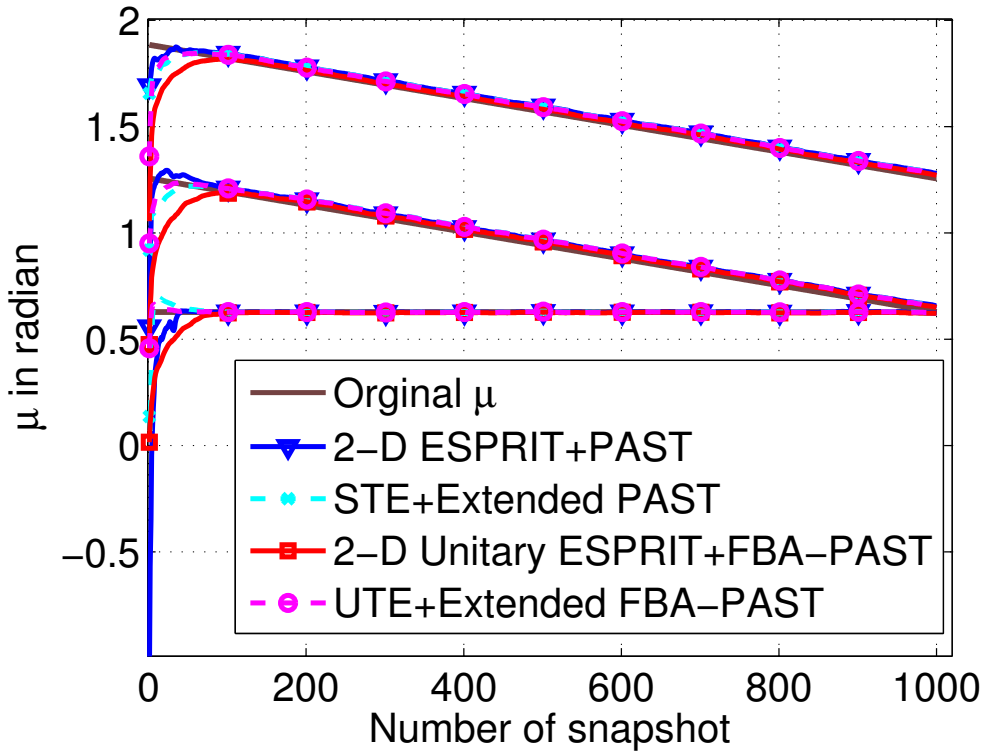


(a) LPA vs. the number of snapshots

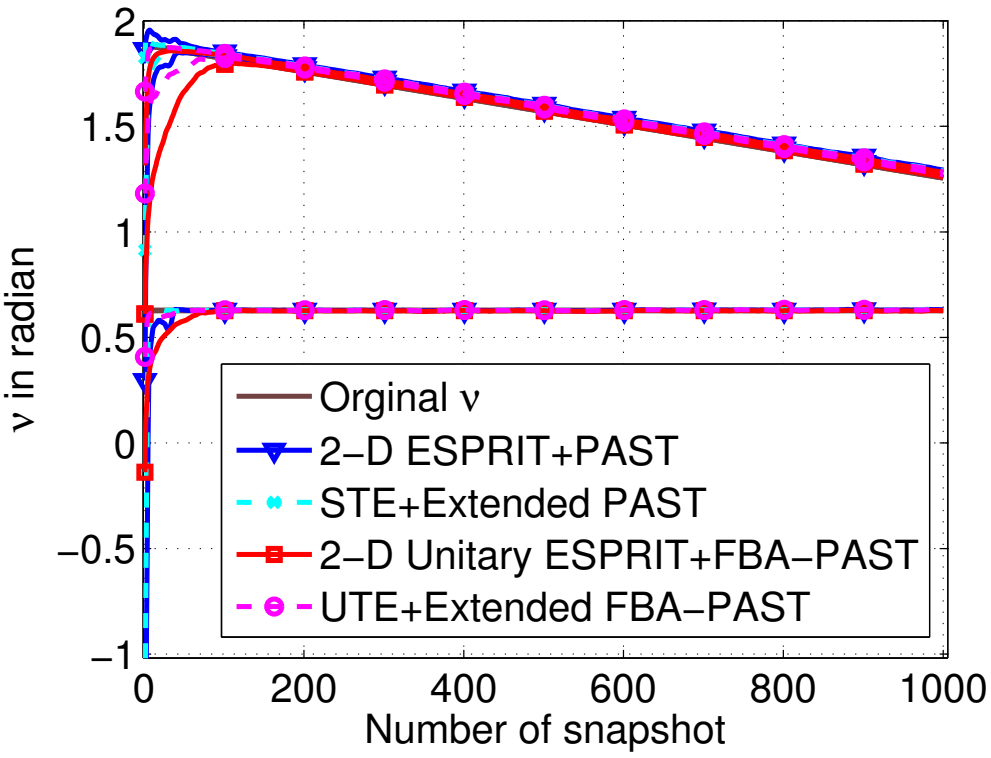


(b) RMSE vs. the number of snapshots

Fig. 3.5: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0$ for scenario 1

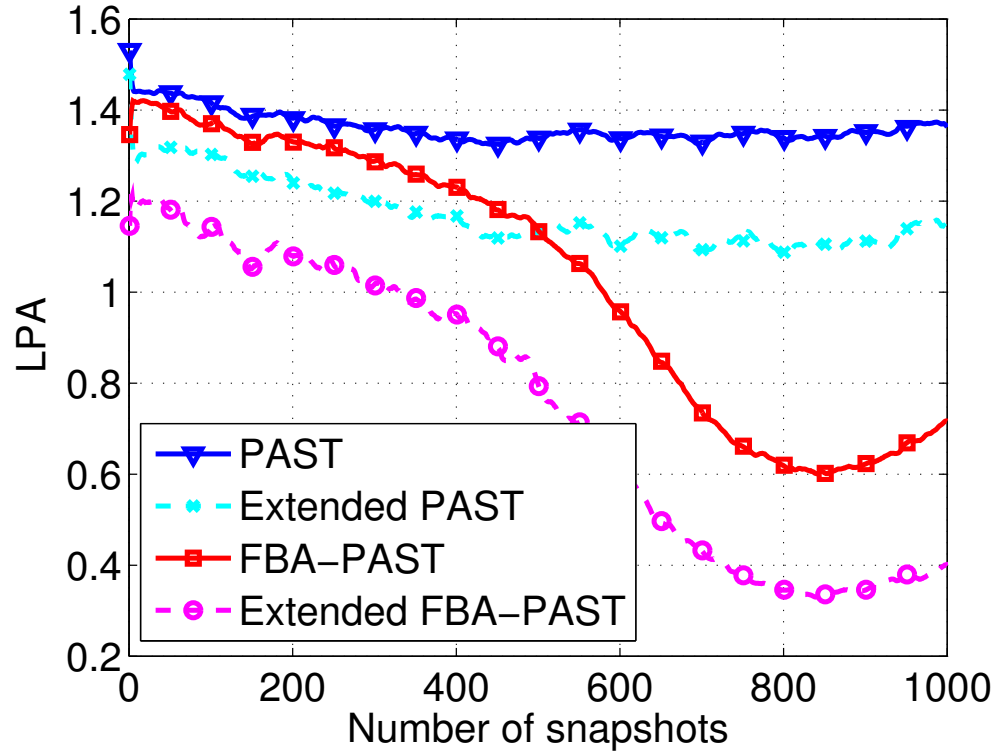


(a) Estimated spatial frequencies on the horizontal plane

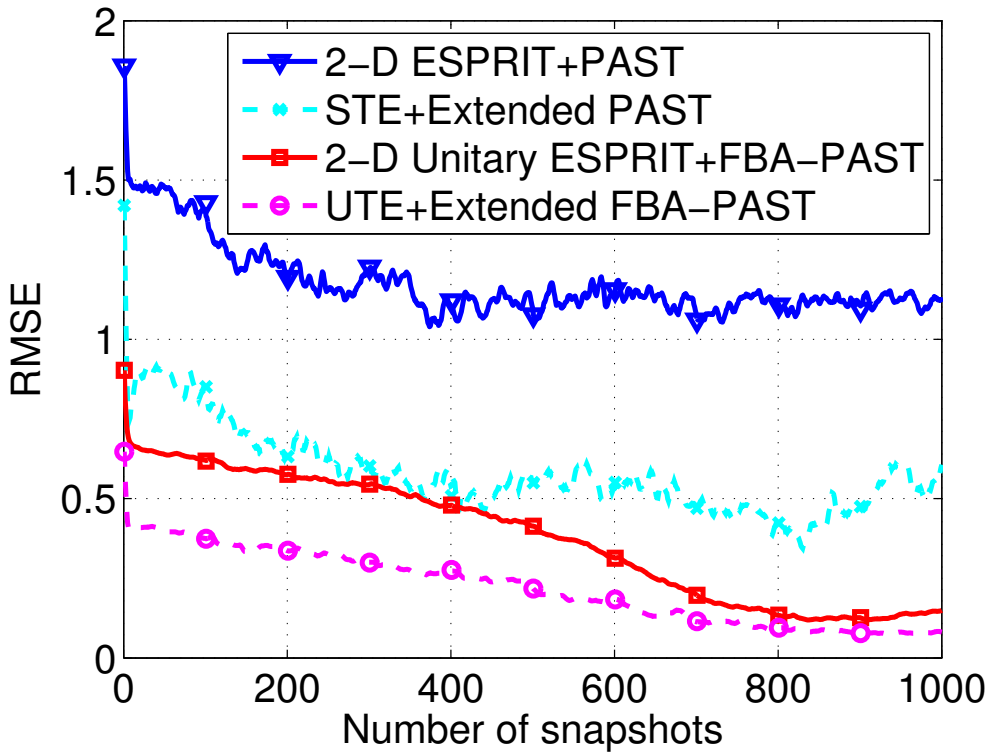


(b) Estimated spatial frequencies on the vertical plane

Fig. 3.6: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0$ for scenario 1

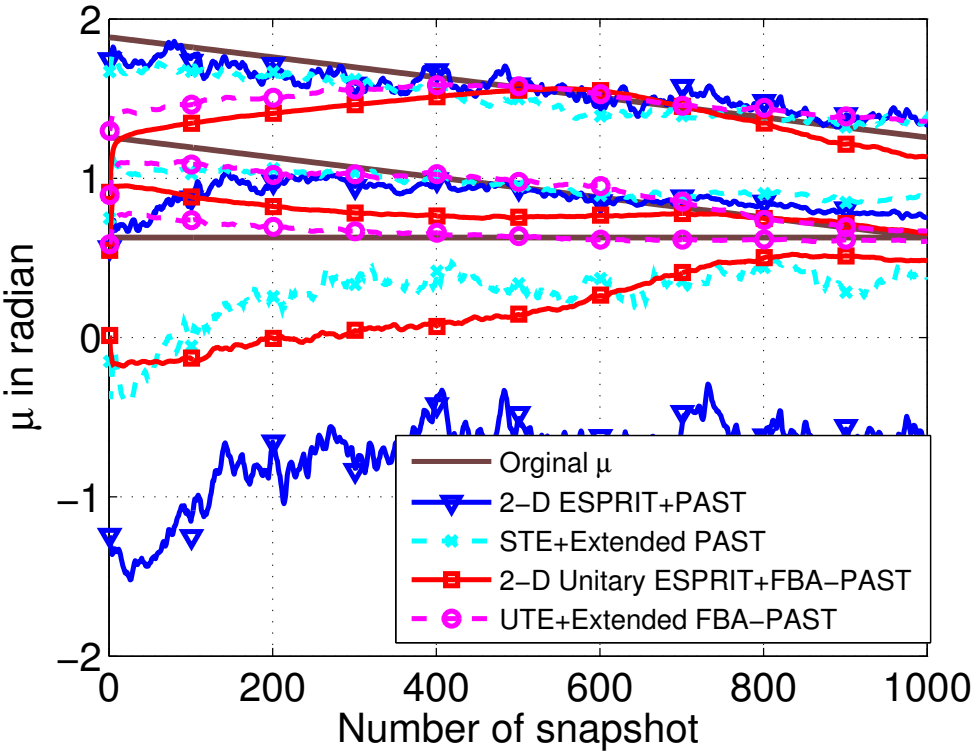


(a) LPA vs. the number of snapshots

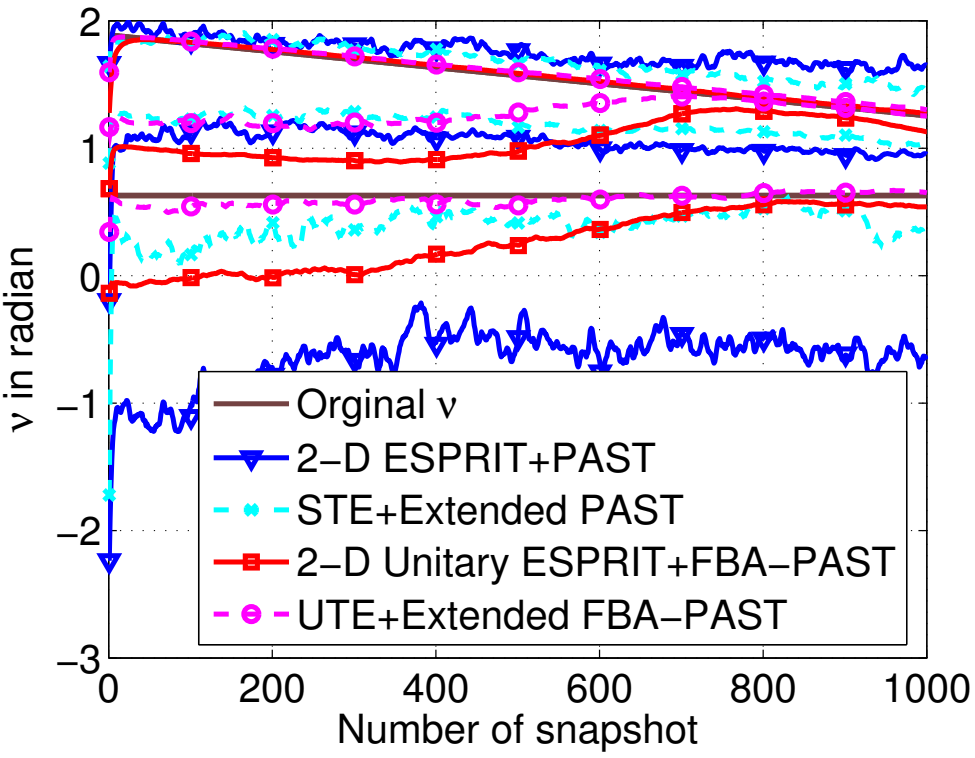


(b) RMSE vs. the number of snapshots

Fig. 3.7: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0.99$ for scenario 1

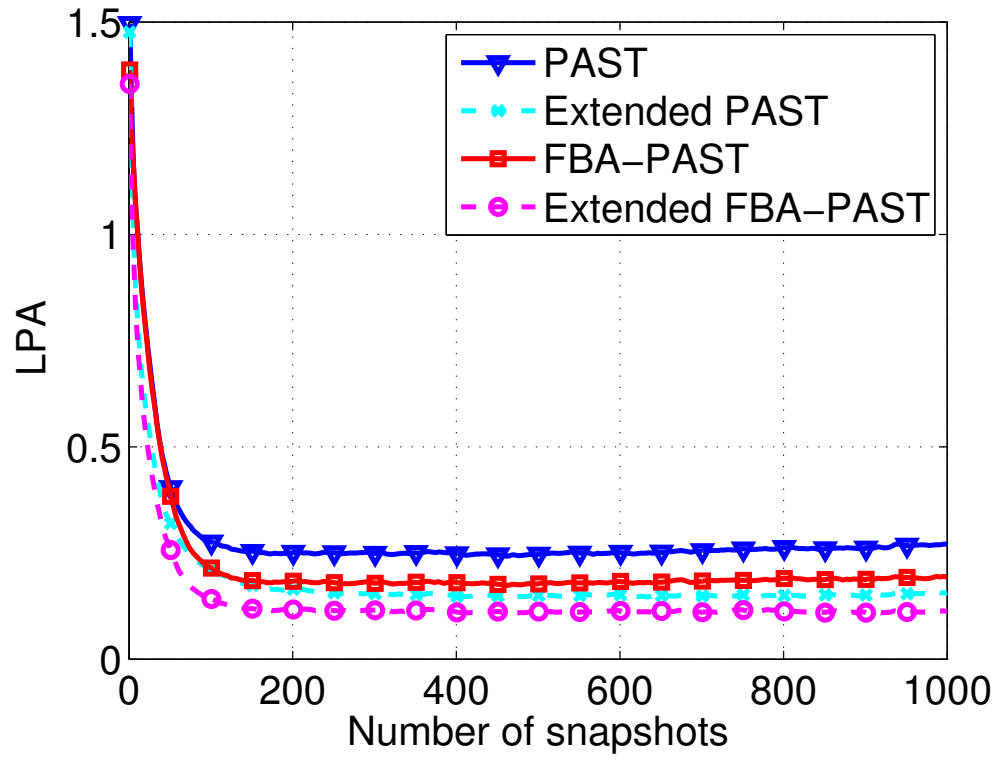


(a) Estimated spatial frequencies on the horizontal plane

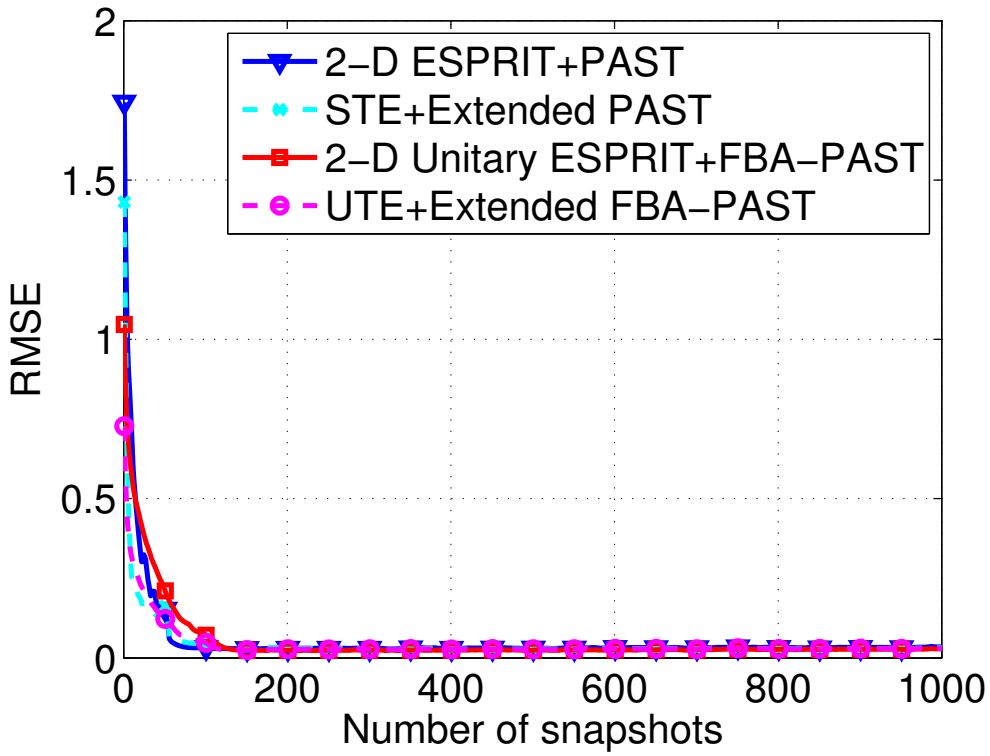


(b) Estimated spatial frequencies on the vertical plane

Fig. 3.8: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0.99$ for scenario 1

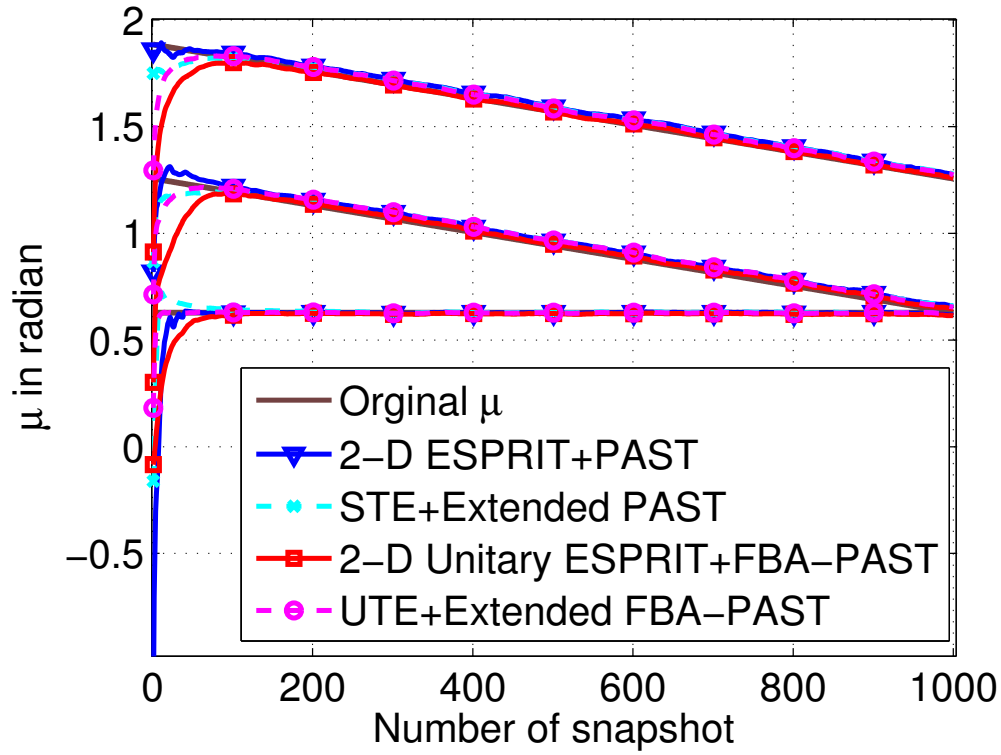


(a) LPA vs. the number of snapshots

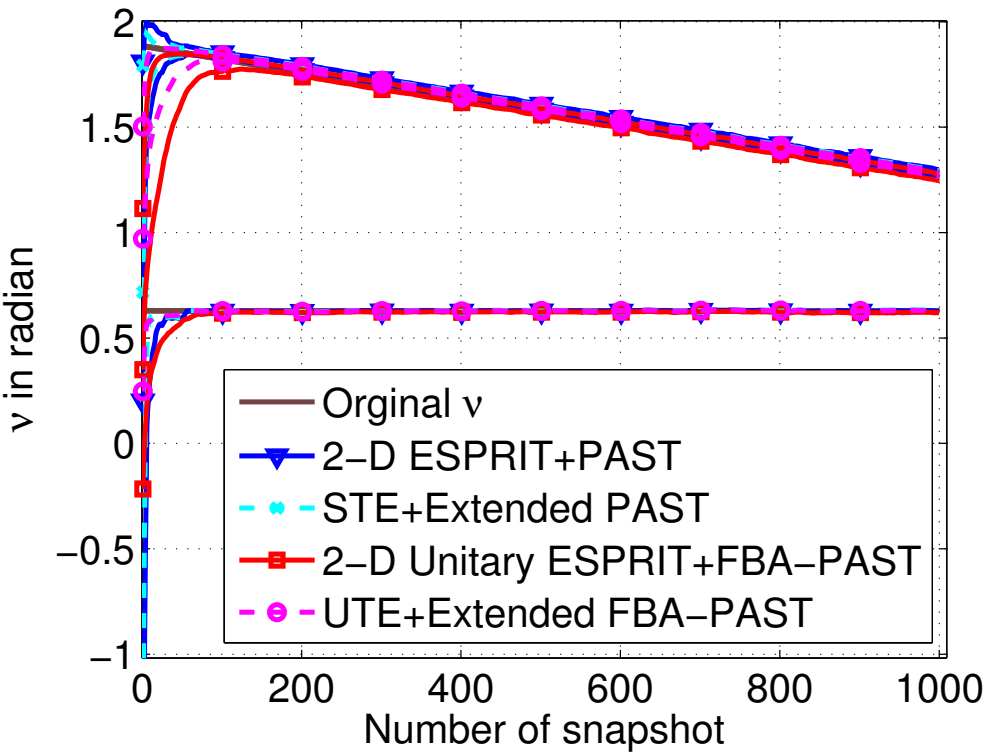


(b) RMSE vs. the number of snapshots

Fig. 3.9: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= -3$ dB and $\rho = 0$ for scenario 1

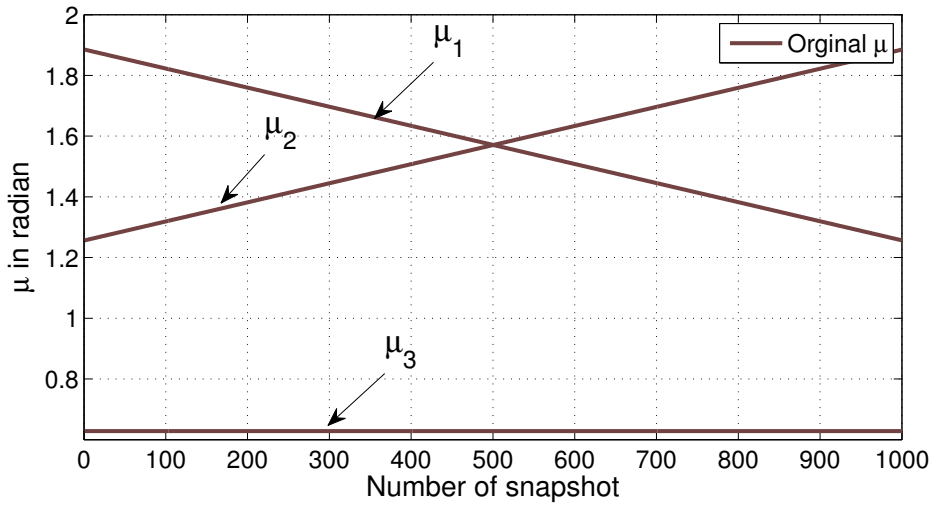


(a) Estimated spatial frequencies on the horizontal plane

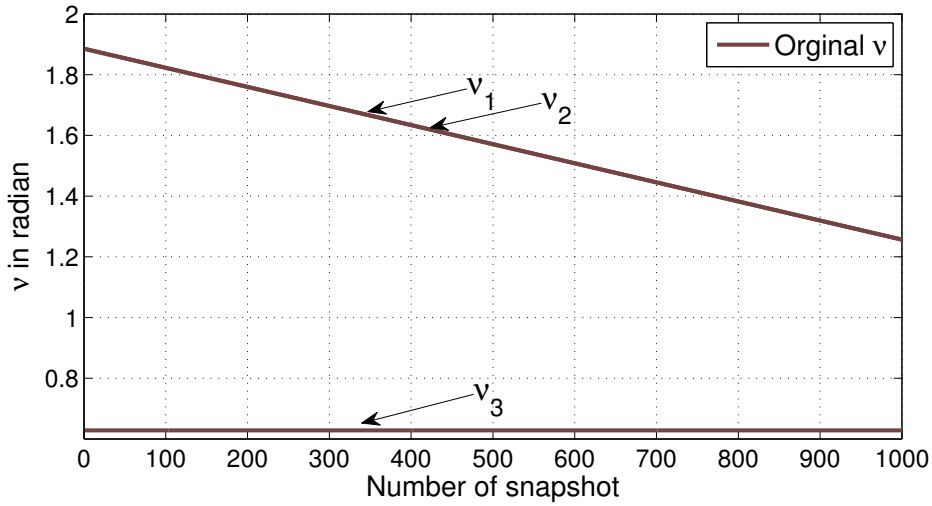


(b) Estimated spatial frequencies on the vertical plane

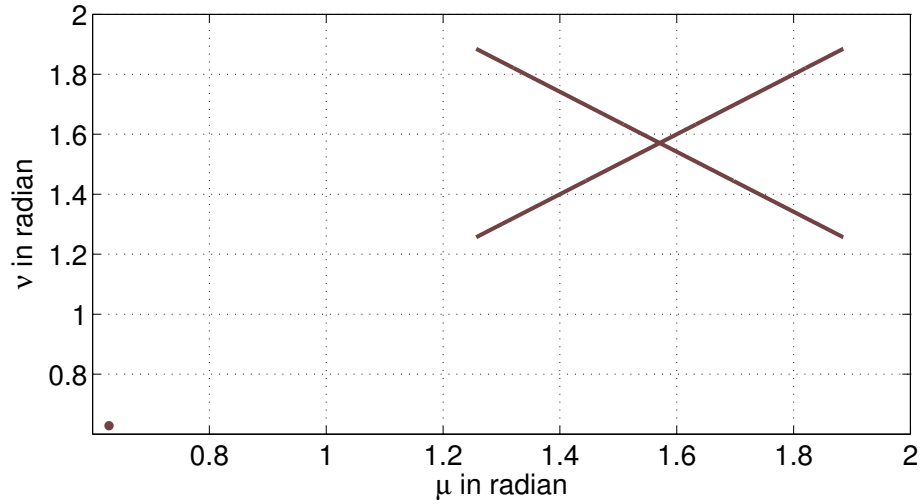
Fig. 3.10: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= -3$ dB and $\rho = 0$ for scenario 1



(a) Spatial frequencies on the horizontal plane

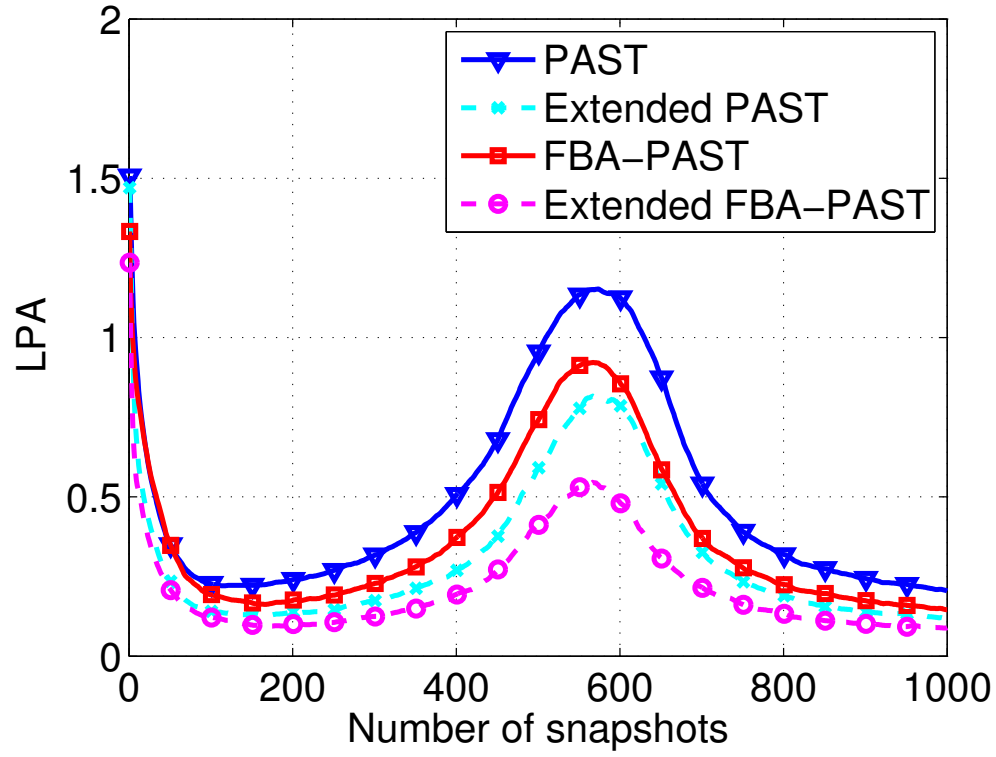


(b) Spatial frequencies on the vertical plane

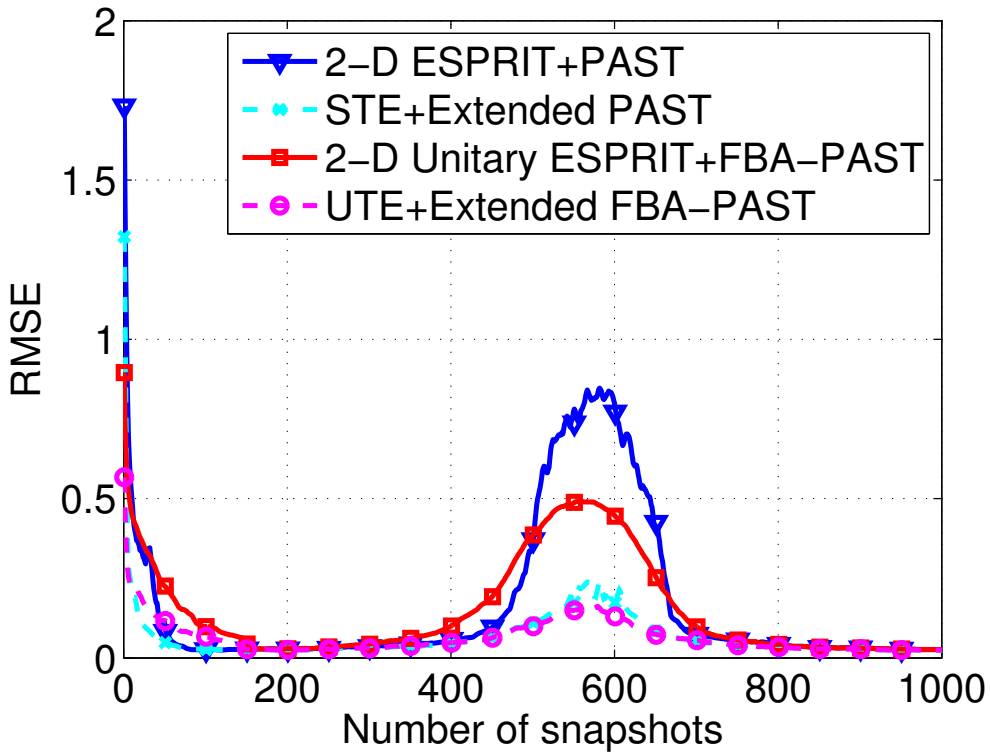


(c) ν vs. μ

Fig. 3.11: Spatial frequencies of 3 sources vs. the number of snapshots for scenario 2

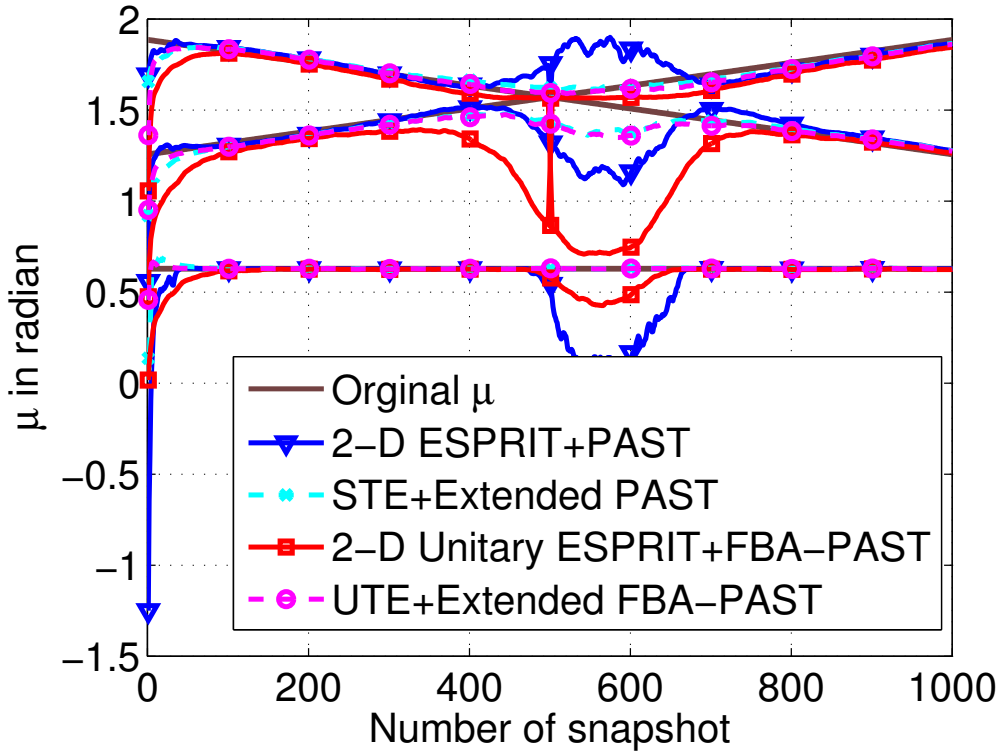


(a) LPA vs. the number of snapshots

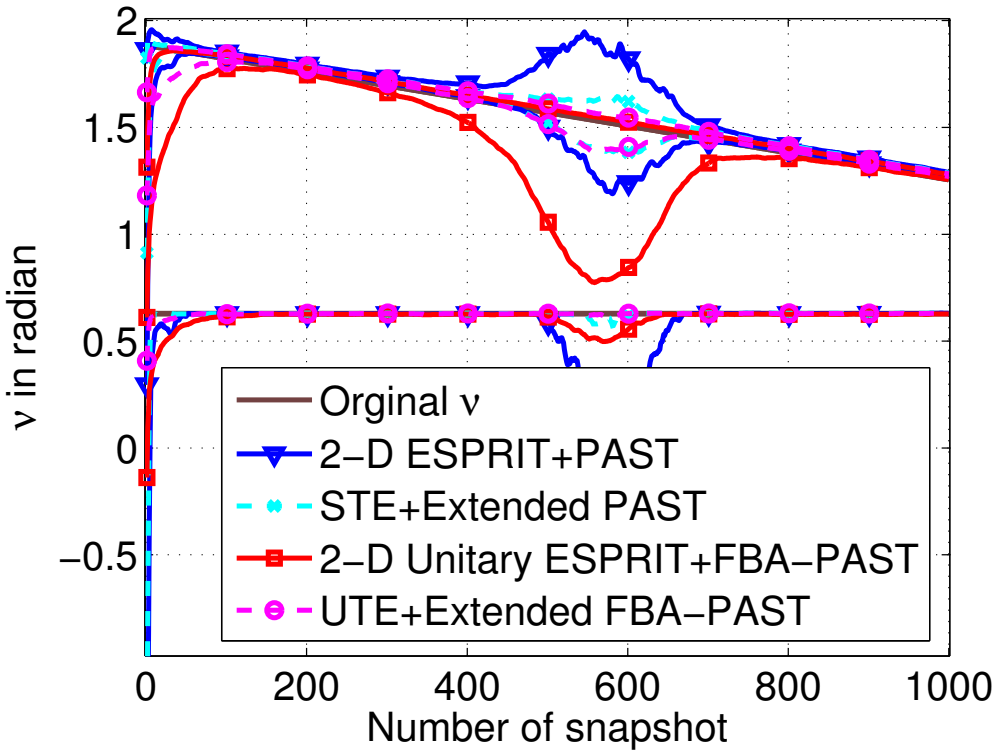


(b) RMSE vs. the number of snapshots

Fig. 3.12: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0$ for scenario 2

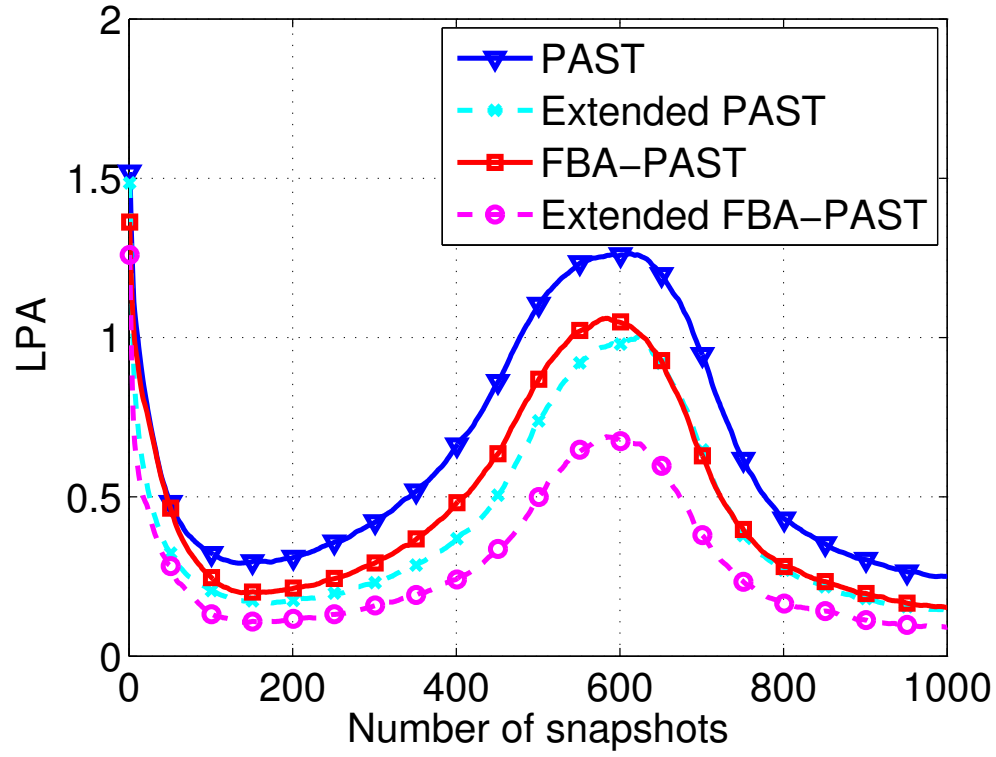


(a) Estimated spatial frequencies on the horizontal plane

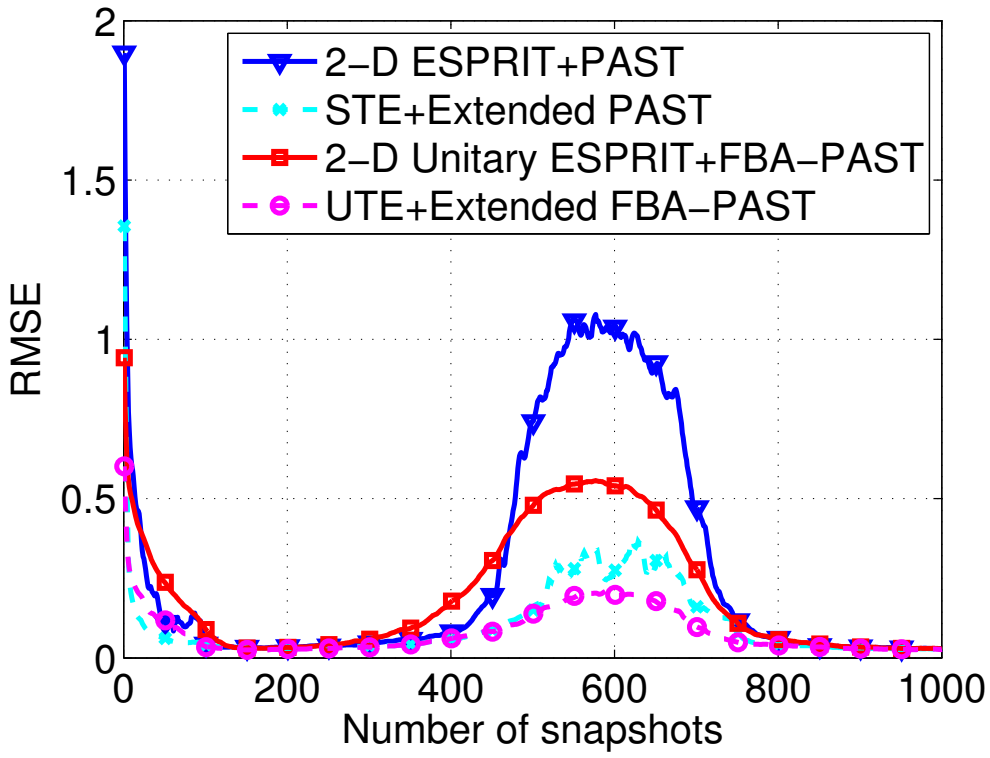


(b) Estimated spatial frequencies on the vertical plane

Fig. 3.13: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0$ for scenario 2

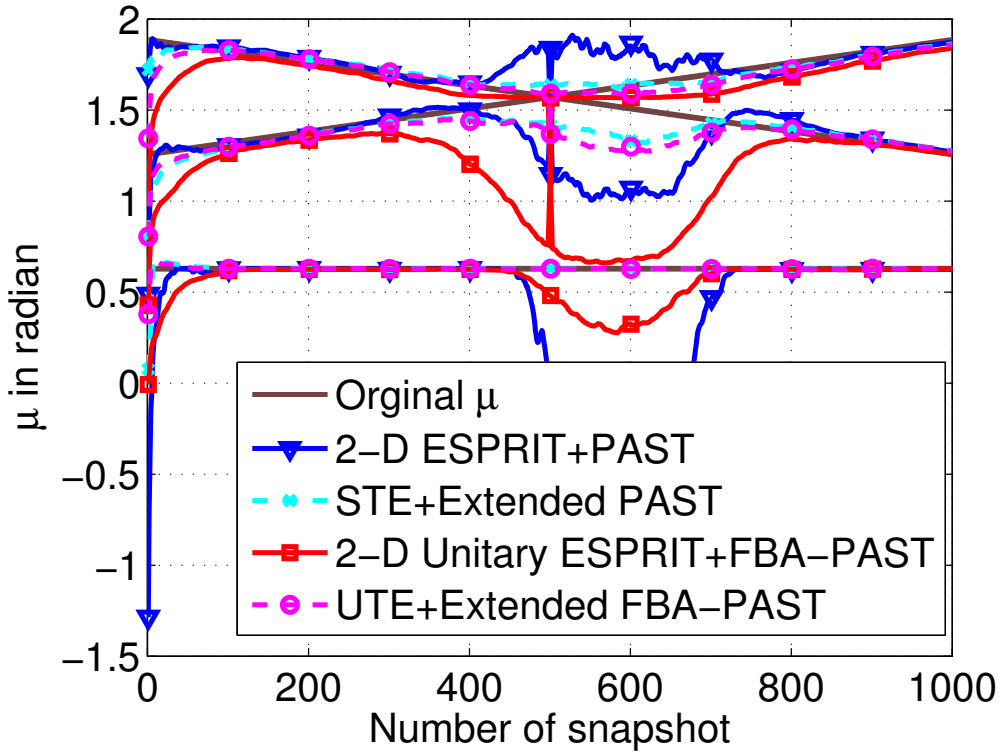


(a) LPA vs. the number of snapshots

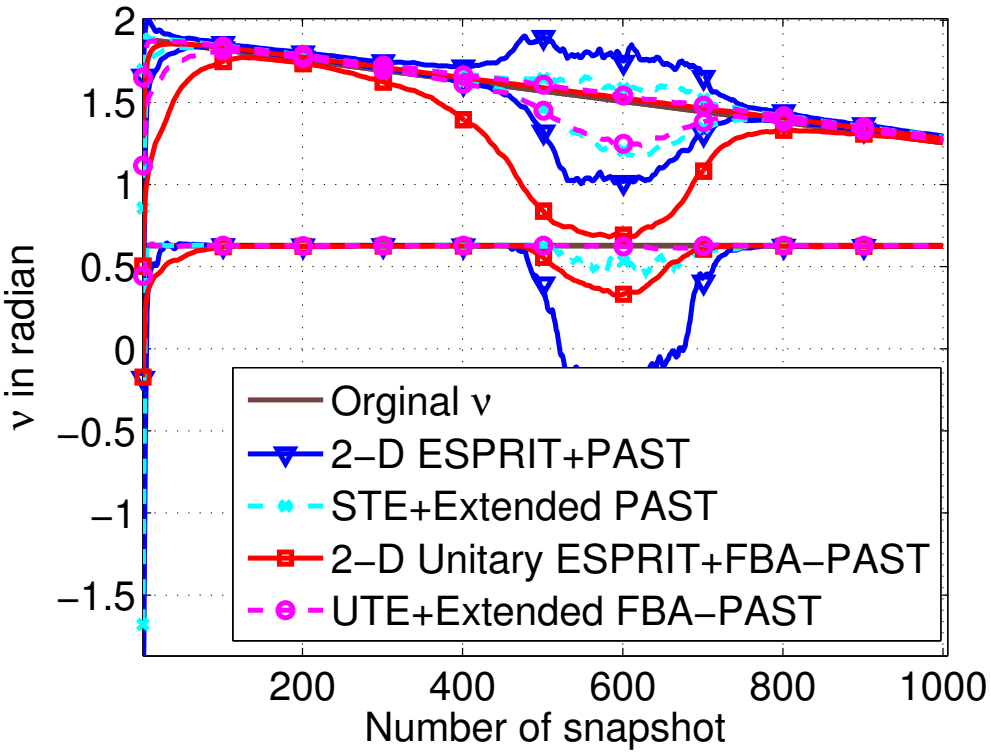


(b) RMSE vs. the number of snapshots

Fig. 3.14: LPA and RMSE vs. the number of snapshots impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0.5$ for scenario 2

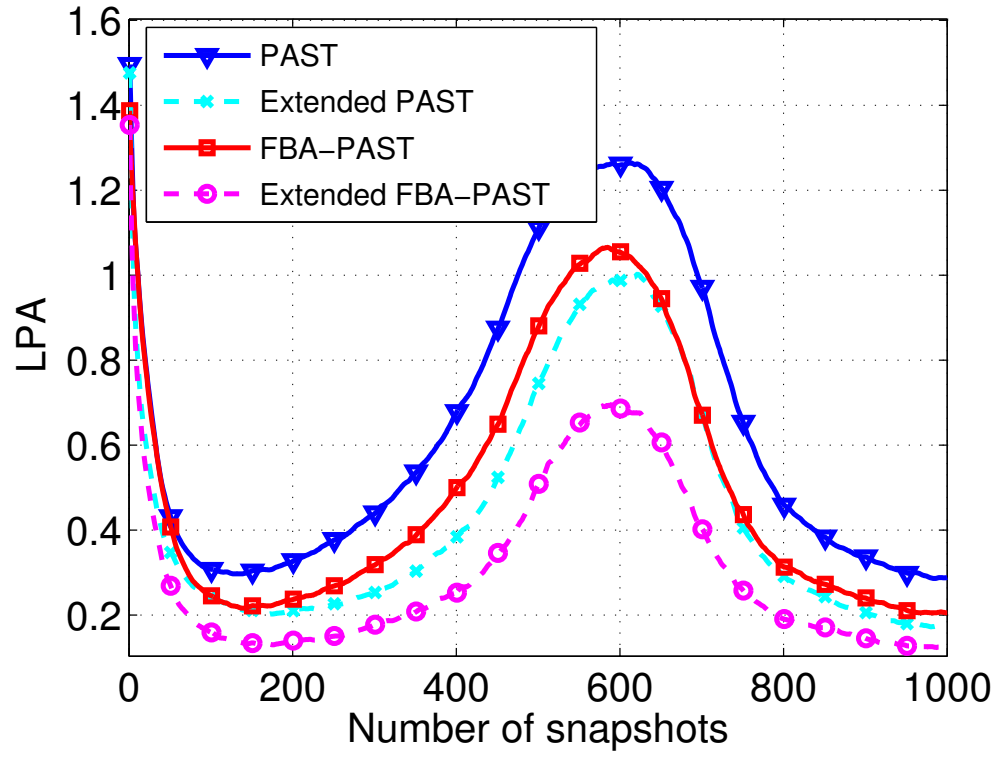


(a) Estimated spatial frequencies on the horizontal plane

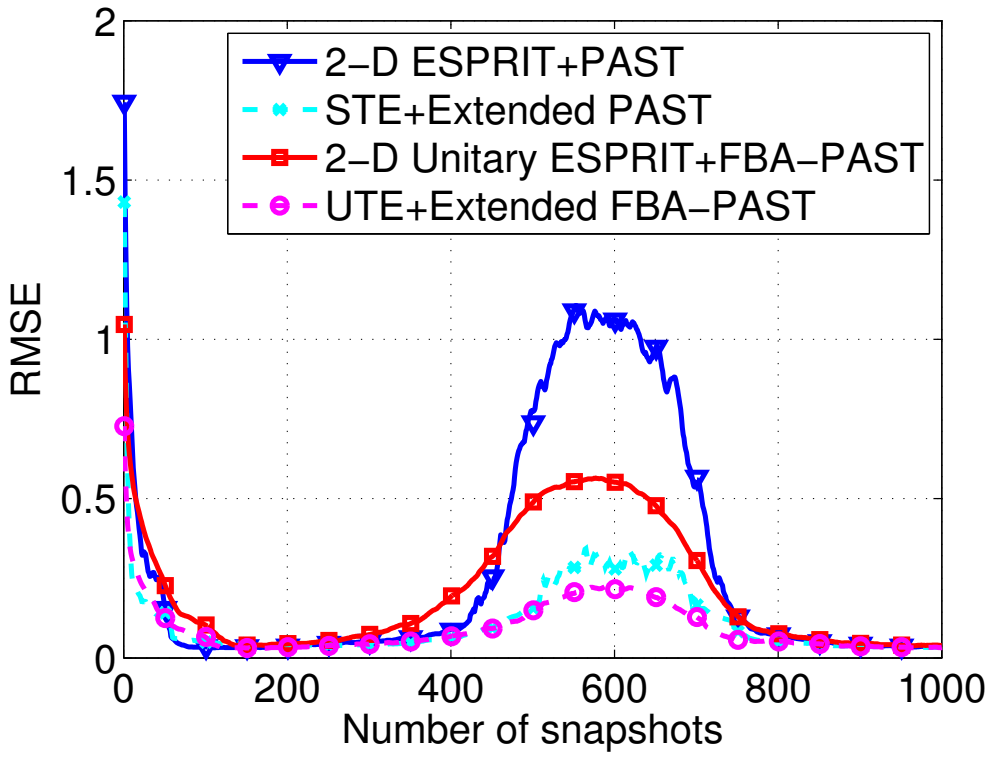


(b) Estimated spatial frequencies on the vertical plane

Fig. 3.15: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0.5$ for scenario 2

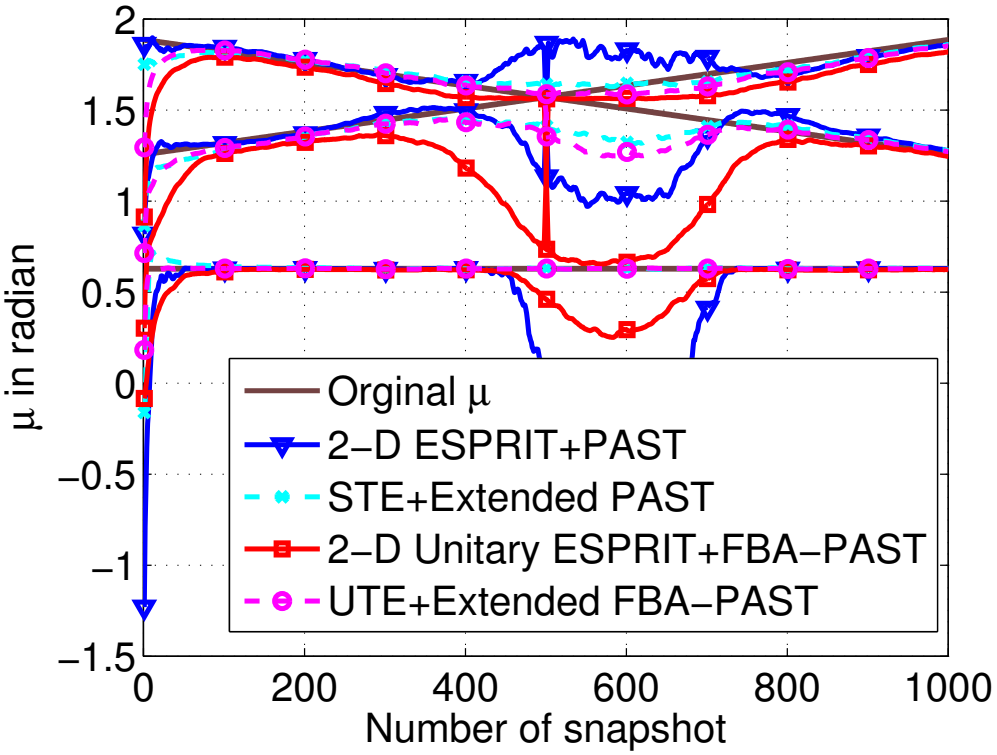


(a) LPA vs. the number of snapshots

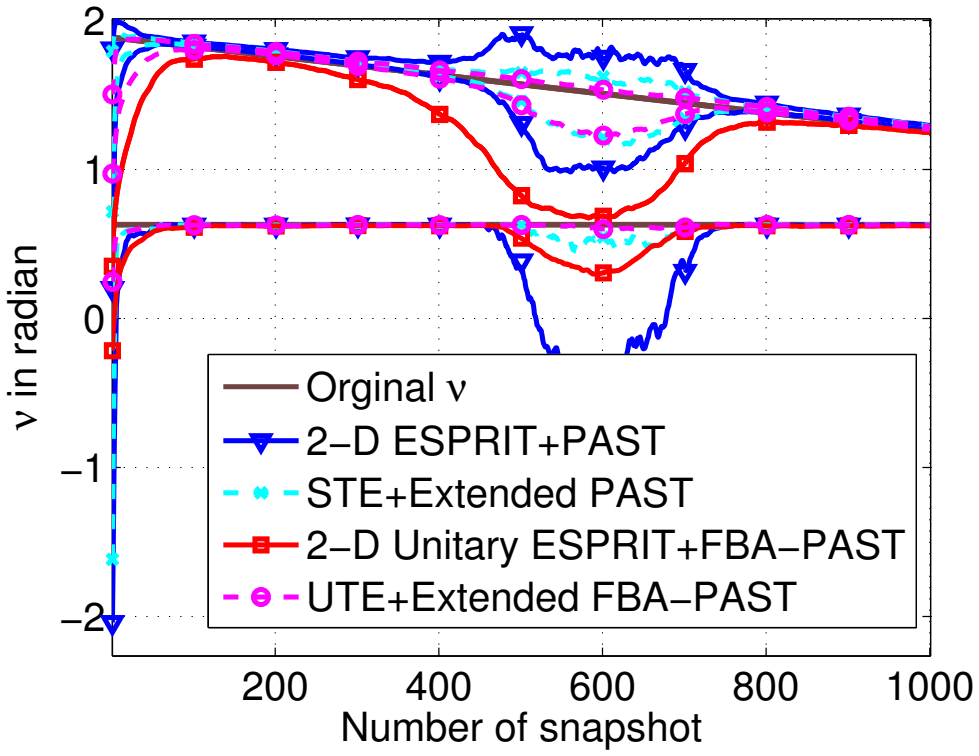


(b) RMSE vs. the number of snapshots

Fig. 3.16: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= -3$ dB and $\rho = 0$ for scenario 2



(a) Estimated spatial frequencies on the horizontal plane



(b) Estimated spatial frequencies on the vertical plane

Fig. 3.17: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= -3$ dB and $\rho = 0$ for scenario 2

4. ADAPTIVE UNITARY ESPRIT AND ADAPTIVE 2-D UNITARY ESPRIT

High-resolution parameter estimation has found its way in a lot of applications, e.g., DOA estimation. Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [RPK86] is one of such algorithms that has lower mathematical complexity comparing to other algorithms such as the Multiple Signal Classification (MUSIC) algorithm [Sch86].

Moreover, an adaptive version of the ESPRIT algorithm has been developed based on the PAST algorithm as shown in [BRD03]. Here, Badeua reduced the mathematical complexity and kept the same performance as good as using ESPRIT algorithm.

In this chapter, we first have a brief review of ESPRIT, Unitary ESPRIT in [HN95], and the Adaptive ESPRIT algorithms based on the PAST subspace tracker [BRD03]. After that, we propose adaptive versions of Unitary ESPRIT and R-D Unitary ESPRIT based on the FBA-PAST algorithm.

At the end, we evaluate the performance of Adaptive 2-D Unitary ESPRIT based on FBA-PAST compared to Adaptive 2-D ESPRIT based on the PAST algorithm.

4.1 Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT)

The ESPRIT algorithm is a high-resolution estimation technique applied to estimate DOA problems [RPK86]. Roy benefited from the feature of the array geometry and took into account a constraint on the structure of it to reduce the computational complexity. This constraint on the structure can be described by a planar array of an arbitrary geometry contains of M sensor doublets and separated by a constant displacement vectors l as depicted in Fig. 4.1 [RPK86]. In other words, we consider that the planar array can be comprised of two subarrays shifting each one with respect to the other by a constant displacement vectors l.

Moreover, the mathematical complexity has been reduced by using the rotational invariance technique as coming soon.

To explain the rotational or translation invariance technique, let us consider a data model of d narrowband sources located in the far field from the sensor array to be given equivalent to (2.24) as

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{W}. \tag{4.1}$$

According to [RPK86], the observed data $\boldsymbol{X}$ can be divided into two doublet sensors as follows

$$\begin{aligned} \boldsymbol{X}_1 &= \tilde{\boldsymbol{A}}\boldsymbol{S} + \boldsymbol{W}_1, \\ \boldsymbol{X}_2 &= \tilde{\boldsymbol{A}}\boldsymbol{\Phi}\boldsymbol{S} + \boldsymbol{W}_2. \end{aligned} \tag{4.2}$$

Here, $\boldsymbol{X}_k \in \mathbb{C}^{M_k \times N_k}, k = 1, 2$ denotes the observation signal at the k-th doublet, $\tilde{\boldsymbol{A}} \in \mathbb{C}^{M_i \times d}$ symbolizes the array steering matrix , $\boldsymbol{S} \in \mathbb{C}^{d \times N}$ representes the impinging wavefronts, $\boldsymbol{\Phi} = \text{diag}\{e^{j\mu_1}, e^{j\mu_2}, ..., e^{j\mu_d}\} \in \mathbb{C}^{d \times d}$ is the phase delays matrix between the doublet sensors for d wavefronts, where $\mu_i, i = 1, ..., d$ symbolize the spatial frequency, and $\boldsymbol{W}_k$ is the additive noise at the k-th doublet with the same size of $\boldsymbol{X}_k$. In order

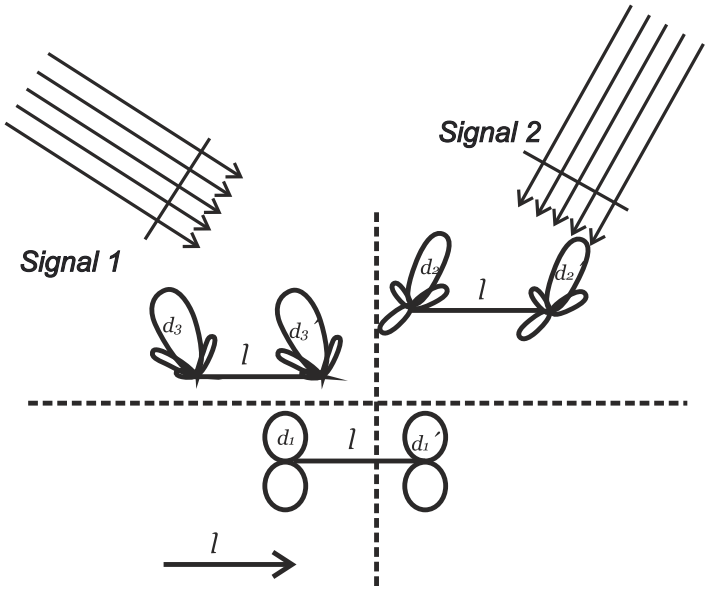


Fig. 4.1: Sensor array geometry for multiple source DOA estimation using ESPRIT

to obtain the relation between the observation sets $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, we compute the SVD of the latter. This yields two sets of signal subspaces $\boldsymbol{U}_{s_1}$ and $\boldsymbol{U}_{s_2}$ that both span the same signal subspace of the array steering matrix $\boldsymbol{A}$. Since $\text{im}\{\boldsymbol{A}\} \approx \text{im}\{\boldsymbol{U}_s\}$, there

must exist a unique nonsingular matrix $\boldsymbol{T}$ such that

$$\boldsymbol{U}_s = \begin{bmatrix} \boldsymbol{U}_{s_1} \\ \boldsymbol{U}_{s_2} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{A}} \\ \tilde{\boldsymbol{A}}\boldsymbol{\Phi} \end{bmatrix} \boldsymbol{T} = \boldsymbol{A}\boldsymbol{T}. \tag{4.3}$$

According to [RPK86], the relation above can yield

$$\mathrm{im}\left\{\boldsymbol{A}\right\} \approx \mathrm{im}\left\{\boldsymbol{U}_{s_1}\right\} \approx \mathrm{im}\left\{\boldsymbol{U}_{s_2}\right\}. \tag{4.4}$$

Since the rows of $\boldsymbol{A}$ related to the elements of sensor array, a particular subarray configuration can be explicated by two selection matrices $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$, respectively. In our study, we consider a Uniform Linear Array (ULA) or URA, and the two selection matrices are chosen to be centro-symmetric with respect to one another according to

$$\boldsymbol{J}_2 = \boldsymbol{\Pi}_m \boldsymbol{J}_1 \boldsymbol{\Pi}_M, \tag{4.5}$$

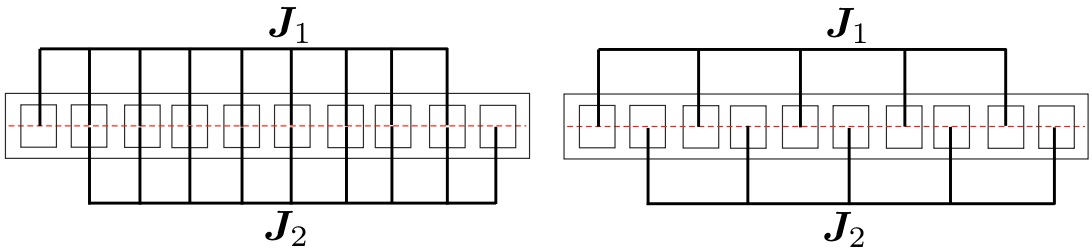
with $\boldsymbol{\Pi}_p$ is the exchange matrix.

To visualize the concept of the selection matrices, let us consider that our sensor array is a ULA. So that we can identify a $\boldsymbol{J}_1$ to pick the first $m = M - 1$ rows of $\boldsymbol{A}$, whereas $\boldsymbol{J}_2$ to select the last $m = M - 1$ rows, i.e., a maximum overlapping case, as visualized in Fig. 4.2(a). Other example is depicted as well in Fig. 4.2(b) for non-overlapping case, so that $\boldsymbol{J}_1$ can pick the first odd rows $m = \frac{M}{2} - 1$ of $\boldsymbol{A}$, whereas $\boldsymbol{J}_2$ selected the even rows, M is an even number in this example.

The shift invariance property of the array steering matrix $\boldsymbol{A}$ can be written such as

$$\boldsymbol{J}_1\boldsymbol{A}\boldsymbol{\Phi} = \boldsymbol{J}_2\boldsymbol{A}, \quad \text{where} \quad \boldsymbol{\Phi} = \mathrm{diag}\{e^{j\mu_i}\}_{i=1}^{d}. \tag{4.6}$$

As we can see from (4.4), the columns of the array steering matrix $\boldsymbol{A}$ approximately



(a) Uniform linear array of $M = 10$ identical sensors with maximum overlapping

(b) Uniform linear array of $M = 10$ identical sensors without overlapping

Fig. 4.2: ULA of $M = 10$ identical sensors. $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ represent the selection matrices

span the same signal subspaces. So that, the shift-invariance property in (4.6) can be given in the term of signal eigenvectors $\boldsymbol{U}_s$ according to (4.3) as

$$\boldsymbol{J}_1\boldsymbol{U}_s\boldsymbol{T}\boldsymbol{\Phi} = \boldsymbol{J}_2\boldsymbol{U}_s\boldsymbol{T}, \longleftrightarrow \boldsymbol{J}_1\boldsymbol{U}_s\boldsymbol{\Psi} = \boldsymbol{J}_2\boldsymbol{U}_s, \quad \text{where} \quad \boldsymbol{\Psi} = \boldsymbol{T}\boldsymbol{\Phi}\boldsymbol{T}^{-1}. \tag{4.7}$$

Since $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi} \in \mathbb{C}^{d\times d}$ are related via eigenvalues preserving transformation, We only need to compute the eigenvalues of the matrix $\boldsymbol{\Psi}$.

For the ULA sensors, the connection between the spatial frequencies μ_i and DOA's θ_i can be obtained via the relationships

$$\mu_i = \arg(\theta_i) \quad \text{and} \quad \theta_i = \arcsin(-\frac{\lambda}{2\pi l} \cdot \mu_i), \quad 1 \leq i \leq d. \tag{4.8}$$

The DOA estimation via ESPRIT algorithm is summarized in Tab. 4.1 where the invariance techniques such as Least Squares (LS), Total least Squares (TLS), and Structured Least Squares (SLS), see [HN96], can be used to solve the invariance equation of (4.7).

1. **Signal Subspace Estimation:** compute $\boldsymbol{U}_s \in \mathbb{C}^{M\times d}$
   - as the d dominant left singular vectors of $\boldsymbol{X} \in \mathbb{C}^{M\times N}$ (square root approach),
   - or the d dominant eigenvectors of $\boldsymbol{X}\boldsymbol{X}^H \in \mathbb{C}^{M\times M}$ (covariance approach).
2. **Solution of the Invariance Equation:** Then solve

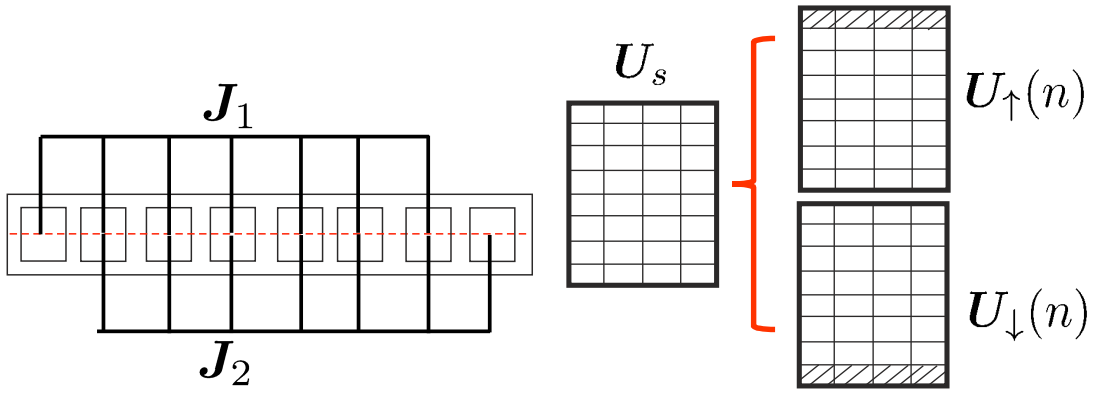
   $$\boldsymbol{J}_1\boldsymbol{U}_s\boldsymbol{\Psi} \approx \boldsymbol{J}_2\boldsymbol{U}_s \in \mathbb{C}^{m\times d}$$

   by means of LS,TLS, or SLS.
3. **Spatial Frequency Estimation:** calculate the eigenvalues of the resulting complex-valued solution

   $$\boldsymbol{\Psi} = \boldsymbol{T}\boldsymbol{\Phi}\boldsymbol{T}^{-1} \in \mathbb{C}^{d\times d} \text{ with } \boldsymbol{\Phi} = \text{diag}\{\phi_i\}_{i=1}^{d}$$

   - $\mu_i = \arg(\phi_i), \quad 1 \leq i \leq d.$

Tab. 4.1: Summary of the standard ESPRIT algorithm



(a) The selection matrices $\boldsymbol{J}_1, \boldsymbol{J}_2 \in \mathbb{C}^{M-1\times M}$ of a ULA with $M = 8$ identical sensors

(b) the abstracted signal subspaces $\boldsymbol{U}_\downarrow(n)$ and $\boldsymbol{U}_\uparrow(n) \in \mathbb{C}^{M-1\times d}$ from the signal subspaces $\boldsymbol{U} \in \mathbb{C}^{M\times d}$, where M is 8 rows

Fig. 4.3: The abstracted signal subspaces according to the selection matrices of maximum overlapping

4.2 Adaptive ESPRIT Algorithm Based on the PAST Subspace Tracker

Adaptive ESPRIT tracks signal parameters based on tracking the signal subspace $\boldsymbol{U}_s(n)$ of the observation data via combining the PAST algorithm and the ESPRIT algorithm in one algorithm as clarified in [BRD03]. As we have shown in the ESPRIT algorithm, the matrix $\boldsymbol{\Phi}$ can be calculated via LS invariance technique at each time step. Thus, at each update, the matrix $\boldsymbol{\Phi}(n)$ requires $O(Md^2)$ operations in addition to $O(d^2)$ operations to extract the spatial frequencies $\mu_i(n)$. Badeau in his paper [BRD03] consummated a fast update in calculating $\boldsymbol{\Phi}(n)$. The complexity of the calculating the DOA has been reduced to $O(Md)$ operations by considering the recursion concept in his algorithm.

The following mathematical derivation steps summarize the Adaptive ESPRIT algorithm based on the PAST subspace tracker. First, let $\boldsymbol{U}_\downarrow(n)$ and $\boldsymbol{U}_\uparrow(n)$ be the matrices extracted from deleting some rows of $\boldsymbol{U}(n)$ according to the selection matrices, $\boldsymbol{J}_1$, $\boldsymbol{J}_2$ respectively. Here, $\boldsymbol{J}_1$, $\boldsymbol{J}_2$ have been chosen to be maximum overlapping matrices.

For instance, Fig. 4.3(a) depicts a ULA of $M = 8$ identical sensors which the selection matrices have been chosen to be centor-symmetric matrices with maximum overlapping. Thus, at each time step, the solution of the invariance equation (4.6) by using LS is written according to [BRD03] as

$$\boldsymbol{\Phi}(n) \;=\; \boldsymbol{U}_\downarrow(n)^\dagger \cdot \boldsymbol{U}_\uparrow(n) \in \mathbb{C}^{d\times d}, \tag{4.9}$$

where the extracted matrices can be computed in the following way,

$$\boldsymbol{U}_{\downarrow}(n) \;=\; \boldsymbol{J}_1 \cdot \boldsymbol{U}(n) = \boldsymbol{U}_{\downarrow}(n-1) + \boldsymbol{e}_{\downarrow}(n) \cdot \boldsymbol{g}(n)^{\mathrm{H}} \in \mathbb{C}^{(M-1)\times d}, \tag{4.10}$$

$$\boldsymbol{U}_{\uparrow}(n) \;=\; \boldsymbol{J}_2 \cdot \boldsymbol{U}(n) = \boldsymbol{U}_{\uparrow}(n-1) + \boldsymbol{e}_{\uparrow}(n) \cdot \boldsymbol{g}(n)^{\mathrm{H}} \in \mathbb{C}^{(M-1)\times d}. \tag{4.11}$$

Here, $\boldsymbol{e}_{\downarrow}(n) = \boldsymbol{J}_1 \cdot \boldsymbol{e}(n)$ and $\boldsymbol{e}_{\uparrow}(n) = \boldsymbol{J}_2 \cdot \boldsymbol{e}(n)$ are in size$(M-1)$ and a gain matrix $\boldsymbol{g}(n)$ is in size d.

To visualize the extracted form of the abstracted signal subspaces according to the selection matrices with maximum overlapping, Fig. 4.3(b) has been given.

Now, We can define a matrix $\boldsymbol{C}(n)$ according to

$$\boldsymbol{C}(n) \;=\; \boldsymbol{U}_{\downarrow}(n)^{\mathrm{H}} \cdot \boldsymbol{U}_{\downarrow}(n) \in \mathbb{C}^{d\times d}. \tag{4.12}$$

As we can see, the matrix $\boldsymbol{C}(n)$ is a Hermitian matrix. So if the invariance of matrix $\boldsymbol{C}(n)$ is non-singular, we can compute the invariance as

$$\boldsymbol{W}(n) \;=\; \boldsymbol{C}(n)^{-1}. \tag{4.13}$$

Therefore, the pseudo-inverse of the abstracted signal subspace can be calculated such as

$$\boldsymbol{U}_{\downarrow}(n)^{\dagger} \;=\; \boldsymbol{W}(n) \cdot \boldsymbol{U}_{\downarrow}(n)^{\mathrm{H}}. \tag{4.14}$$

Now, in order to obtain the invariance of $\boldsymbol{C}(n)$, we substitute (4.10) into (4.12) as follows

$$\boldsymbol{C}(n) \;=\; \boldsymbol{C}(n-1) + \boldsymbol{F}(n) \cdot \boldsymbol{L}(n) \cdot \boldsymbol{F}(n)^{\mathrm{H}}, \tag{4.15}$$

where $\boldsymbol{F}(n) \in \mathbb{R}^{d\times 2}$ has been computed as

$$\boldsymbol{F}(n) \;=\; [\boldsymbol{U}_{\downarrow}(n-1)^{\mathrm{H}} \cdot \boldsymbol{e}_{\downarrow}(n) \mid \boldsymbol{g}(n)], \tag{4.16}$$

and the non-singular matrix $\boldsymbol{L}(n) \in \mathbb{R}^{2\times 2}$ can be calculated as

$$\boldsymbol{L}(n) \;=\; \left[\begin{array}{c|c} 0 & 1 \\ \hline 1 & \|\boldsymbol{e}_{\downarrow}(n)\|^2 \end{array}\right]. \tag{4.17}$$

In order to compute the inversion of the formula (4.14), [BRD03] used the Woodbury

identity (3.17). Then, the inversion of equation (4.14) can be calculated as

$$\boldsymbol{W}(n) \;=\; \boldsymbol{W}(n-1) - \boldsymbol{S}(n)\cdot\boldsymbol{\Lambda}(n)\cdot\boldsymbol{S}(n)^{\mathrm{H}}, \tag{4.18}$$

where $\boldsymbol{S}(n)$ is a $d \times 2$ matrix

$$\boldsymbol{S}(n) \;=\; \boldsymbol{W}(n-1)\cdot\boldsymbol{F}(n), \tag{4.19}$$

and $\boldsymbol{\Lambda}(n)$ is a 2×2 matrix

$$\boldsymbol{\Lambda}(n) \;=\; (\boldsymbol{L}(n)^{-1} + \boldsymbol{S}(n)^{\mathrm{H}}\cdot\boldsymbol{F}(n))^{-1}. \tag{4.20}$$

After a few mathematical derivations, pseudo-inverse of the abstracted signal subspace $\boldsymbol{U}_{\downarrow}(n)$ can be calculated with respect to the previous moment as

$$\boldsymbol{U}_{\downarrow}(n)^{\dagger} \;=\; \boldsymbol{U}_{\downarrow}(n-1)^{\dagger} + \boldsymbol{R}(n)\cdot\boldsymbol{T}(n)^{\mathrm{H}}, \tag{4.21}$$

where $\boldsymbol{R}(n)$ is a $d \times 2$ matrix

$$\boldsymbol{R}(n) \;=\; \boldsymbol{S}(n)\cdot\boldsymbol{\Lambda}(n), \tag{4.22}$$

and $\boldsymbol{T}(n)$ is a $(M-1) \times 2$ matrix,

$$\boldsymbol{T}(n) \;=\; \boldsymbol{e}_{\downarrow}(n)[1 \mid 0] - \boldsymbol{U}_{\downarrow}(n-1)\cdot\boldsymbol{S}(n). \tag{4.23}$$

Now, the update of matrix $\boldsymbol{\Phi}(n)$ can be obtained by substituting each of the equations (4.21) and (4.11) into (4.9) as

$$\boldsymbol{\Phi}(n) \;=\; \boldsymbol{\Phi}(n-1) + \boldsymbol{Z}(n)\cdot\boldsymbol{Q}(n)^{\mathrm{H}}, \tag{4.24}$$

where $\boldsymbol{Z}(n)$ is a $d \times 3$ matrix

$$\boldsymbol{Z}(n) \;=\; [\boldsymbol{U}_{\downarrow}(n-1)^{\dagger}\cdot\boldsymbol{e}_{\uparrow}(n) \mid \boldsymbol{R}(n)], \tag{4.25}$$

and $\boldsymbol{Q}(n)$ is a $d \times 3$ matrix

$$\boldsymbol{Q}(n) \;=\; [\boldsymbol{g}(n) \mid \boldsymbol{U}_{\uparrow}(n)^{\mathrm{H}}\cdot\boldsymbol{T}(n)]. \tag{4.26}$$

The pseudo-code of the adaptive ESPRIT algorithm based on PAST algorithm has been summarized in Tab. 4.2.

In the next, we generalize the Adaptive algorithm to a general case of selection matrices by replace the indices of the extracted signal subspaces $\boldsymbol{U}_{\downarrow}(n)$ and $\boldsymbol{U}_{\uparrow}(n) \in \mathbb{C}^{m\times d}$ with $\boldsymbol{U}_1(n)$ and $\boldsymbol{U}_2(n) \in \mathbb{C}^{m\times d}$, $m \leq (M-1)$, respectively .

1. **PAST initialization:** (*cf.* Tab. 3.1) $\boldsymbol{U}_s(0) = \boldsymbol{I}_{M\times d},\ \boldsymbol{P}(0) = \boldsymbol{I}_d$
2. **Adaptive ESPRIT initialization:** $\boldsymbol{W}(0) = \boldsymbol{I}_d$, $\boldsymbol{\Phi}(0) = \boldsymbol{U}_{\downarrow}(0)^{\dagger}\boldsymbol{U}_{\uparrow}(0)$ for $n = 1, 2, ...$
   - **PAST main section:** (*cf.* Tab. 3.1)
   - **ESPRIT main section:**

$$
\begin{aligned}
\boldsymbol{F}(n)^{-1} &= [\boldsymbol{U}_{\downarrow}(n-1)^{\mathrm{H}} \cdot \boldsymbol{e}_{\downarrow}(n) \mid \boldsymbol{g}(n)] \\
\boldsymbol{S}(n) &= \boldsymbol{W}(n-1) \cdot \boldsymbol{F}(n) \\
\boldsymbol{L}(n) &= \left[\begin{array}{c|c} -\|\boldsymbol{e}_{\downarrow}(n)\|^2 & 1 \\ \hline 1 & 0 \end{array}\right] \\
\boldsymbol{\Lambda}(n) &= (\boldsymbol{L}(n)^{-1} + \boldsymbol{S}(n)^{\mathrm{H}} \cdot \boldsymbol{F}(n))^{-1} \\
\boldsymbol{R}(n) &= \boldsymbol{S}(n) \cdot \boldsymbol{\Lambda}(n) \\
\boldsymbol{W}(n) &= \boldsymbol{W}(n-1) - \boldsymbol{S}(n) \cdot \boldsymbol{\Lambda}(n) \cdot \boldsymbol{S}(n)^{\mathrm{H}} \\
\boldsymbol{T}(n) &= \boldsymbol{e}_{\downarrow}(n)[1 \mid 0] - \boldsymbol{U}_{\downarrow}(n-1) \cdot \boldsymbol{S}(n) \\
\boldsymbol{Z}(n) &= [\boldsymbol{U}_{\downarrow}(n-1)^{\dagger} \cdot \boldsymbol{e}_{\uparrow}(n) \mid \boldsymbol{R}(n)] \\
\boldsymbol{Q}(n) &= [\boldsymbol{g}(n) \mid \boldsymbol{U}_{\uparrow}(n)^{\mathrm{H}} \cdot \boldsymbol{T}(n)] \\
\boldsymbol{U}_{\downarrow}(n)^{\dagger} &= \boldsymbol{U}_{\downarrow}(n-1)^{\dagger} + \boldsymbol{R}(n) \cdot \boldsymbol{T}(n)^{\mathrm{H}} \\
\boldsymbol{\Phi}(n) &= \boldsymbol{\Phi}(n-1) + \boldsymbol{Z}(n) \cdot \boldsymbol{Q}(n)^{\mathrm{H}}
\end{aligned}
$$

   - $\{\mu_i\}_{1\leq i\leq d} = \arg(\ \mathrm{eig}(\Phi(n)))$

Tab. 4.2: Summary of Adaptive ESPRIT based on PAST algorithm

4.3 Unitary ESPRIT

The Unitary ESPRIT algorithm is one step development of ESPRIT based on the fact that the phase delay between the two subarrays is unitary and if centro-symmetric array configurations are used [HN95]. Unitary ESPRIT has been formulated in the term of real-valued processing leading to a lower mathematical complexity and a higher estimation accuracy compared to the ESPRIT algorithm.

To clarify this concept, we first consider that the data model are given according

1. **Signal Subspace Estimation:** compute $\boldsymbol{E}_s \in \mathbb{R}^{M\times d}$
   - as the d dominant left singular vectors of $\varphi(\boldsymbol{Z}) \in \mathbb{R}^{M\times N}$ (square root approach),
   - or the d dominant eigenvectors of $\varphi(\boldsymbol{Z})\varphi(\boldsymbol{Z})^H \in \mathbb{R}^{M\times M}$ (covariance approach).
2. **Solution of the Invariance Equation:** Then solve

$$\boldsymbol{K}_1\boldsymbol{E}_s\boldsymbol{\Upsilon} \approx \boldsymbol{K}_2\boldsymbol{E}_s \in \mathbb{R}^{m\times d}$$

   by means of LS,TLS, or SLS.
3. **Spatial Frequency Estimation:** calculate the eigenvalues of the resulting complex-valued solution

$$\boldsymbol{\Upsilon} = \boldsymbol{T}\boldsymbol{\Omega}\boldsymbol{T}^{-1} \in \mathbb{R}^{d\times d} \quad \text{with} \quad \boldsymbol{\Omega} = diag\{\omega_i\}_{i=1}^{d}$$

   - **Reliability test:** If all eigenvalues ω_i are real, the estimates will be reliable. Otherwise, start again with more or more reliable measurements.
   - $\mu_i = 2\text{arctan}(\omega_i), \quad 1 \le i \le d.$

Tab. 4.3: Summary of the Unitary ESPRIT algorithm

to (2.24). Then, we use the extension method according to (3.20). After that, the mapping matrix $\varphi(\boldsymbol{Z}) \in \mathbb{R}^{M\times N}$ can be obtained by using (3.21).

By calculating the SVD of the mapping data, the relation between the real-valued and the complex-valued signal subspaces can be represented according to [HN95] as

$$\boldsymbol{U}_s = \boldsymbol{Q}_M\boldsymbol{E}_s, \tag{4.27}$$

where $\boldsymbol{Q}_M \in \mathbb{C}^{M\times M}$ denotes an arbitrary unitary left $\boldsymbol{\Pi}$-real matrix. Now, by substituting the relation (4.27) into (4.6) and performing a suitable manipulation we obtain

$$\boldsymbol{K}_1\boldsymbol{E}_s\boldsymbol{\Upsilon} \approx \boldsymbol{K}_2\boldsymbol{E}_s, \quad \text{where} \quad \boldsymbol{\Upsilon} = \boldsymbol{T}\boldsymbol{\Omega}\boldsymbol{T}^{-1}, \tag{4.28}$$

with $\boldsymbol{\Omega} = \text{diag}\{\omega_i\}_{i=1}^{d}$ is a diagonal matrix, and $\boldsymbol{K}_1, \boldsymbol{K}_2$ can be equal the real and

imaginary part of $2.\boldsymbol{Q}_m^H \boldsymbol{J}_2 \boldsymbol{Q}_M$ reapectivlay as

$$\begin{aligned} \boldsymbol{K}_1 &= \boldsymbol{Q}_m(\boldsymbol{J}_1 + \boldsymbol{J}_2)\boldsymbol{Q}_M = 2 \cdot \mathrm{Re}\{\boldsymbol{Q}_m \boldsymbol{J}_2 \boldsymbol{Q}_M\} \\ \boldsymbol{K}_2 &= \boldsymbol{Q}_m j(\boldsymbol{J}_1 - \boldsymbol{J}_2)\boldsymbol{Q}_M = 2 \cdot \mathrm{Im}\{\boldsymbol{Q}_m \boldsymbol{J}_2 \boldsymbol{Q}_M\}. \end{aligned} \tag{4.29}$$

Since $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Omega} \in \mathbb{R}^{d \times d}$ are related via eigenvalues preserving transformation, We only need to compute the eigenvalues of the matrix $\boldsymbol{\Omega}$.

We summarize the steps of Unitary ESPRIT algorithm in Tab. 4.3, where the invariance techniques such as LS, TLS, or SLS can be used to solve the invariance equation of (4.28). Note that, the spatial frequency μ_i can be calculated for a ULA scenario as

$$\mu_i = 2\mathrm{arctan}(\omega_i), \quad 1 \leq i \leq d \tag{4.30}$$

4.4 Adaptive Unitary ESPRIT Algorithm Based on the FBA-PAST Subspace Tracker

We propose an adaptive version of Unitary ESPRIT to track the real-valued signal subspaces depending on FBA-PAST algorithm. Our contribution is to reduce the the complexity to $O(Md)$ operations in the real domain, as well as to increase the signal parameters tracking accuracy.

The new adaptive version can be obtained after a few mathematical modification on the Adaptive ESPRIT algorithm as we will describe soon. First, we generalize a general case of selection matrices $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ and calculate $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ according to (4.2) and (4.29). Then, we solve the invariance of (4.28) based on FBA-PAST similar to [BRD03] as

$$\boldsymbol{\Upsilon}(n) = \boldsymbol{E}_1(n)^{\dagger} \cdot \boldsymbol{E}_2(n) \in \mathbb{R}^{d \times d}, \tag{4.31}$$

where the extracted matrices can be computed in the following way

$$\boldsymbol{E}_1(n) = \boldsymbol{K}_1 \cdot \boldsymbol{E}(n) = \boldsymbol{E}_1(n-1) + \boldsymbol{B}_1(n) \cdot \boldsymbol{G}(n)^{\mathrm{H}} \in \mathbb{R}^{m \times d}, \tag{4.32}$$

$$\boldsymbol{E}_2(n) = \boldsymbol{K}_2 \cdot \boldsymbol{E}(n) = \boldsymbol{E}_2(n-1) + \boldsymbol{B}_2(n) \cdot \boldsymbol{G}(n)^{\mathrm{H}} \in \mathbb{R}^{m \times d}, \tag{4.33}$$

where $\boldsymbol{B}_1(n) = \boldsymbol{K}_1 \cdot \boldsymbol{B}(n)$, and $\boldsymbol{B}_2(n) = \boldsymbol{K}_2 \cdot \boldsymbol{B}(n)$ are in size $m \times 2$. Now, by

considering $\boldsymbol{C}(n)$ to be $d \times d$ Hermitian matrix according to

$$\boldsymbol{C}(n) \;=\; \boldsymbol{E}_1(n)^{\mathrm{H}} \cdot \boldsymbol{E}_1(n). \tag{4.34}$$

The invariance of non-singular matrix $\boldsymbol{C}(n)$ can be given as

$$\boldsymbol{W}(n) \;=\; \boldsymbol{C}(n)^{-1} = (\boldsymbol{E}_1(n)^{\mathrm{H}} \cdot \boldsymbol{E}_1(n))^{-1}. \tag{4.35}$$

Therefore the pseudo-inverse of the abstracted signal subspace can be calculated such as

$$\boldsymbol{E}_1(n)^{\dagger} \;=\; \boldsymbol{W}(n) \cdot \boldsymbol{E}_1(n)^{\mathrm{H}}. \tag{4.36}$$

By substituting (4.32) into (4.34), we obtain

$$\boldsymbol{C}(n) \;=\; \boldsymbol{C}(n-1) + \boldsymbol{F}(n) \cdot \boldsymbol{L}(n) \cdot \boldsymbol{F}(n)^{\mathrm{H}}, \tag{4.37}$$

where $\boldsymbol{F}(n) \in \mathbb{R}^{d \times 4}$ has been computed as

$$\boldsymbol{F}(n) \;=\; [\boldsymbol{E}_1(n-1)^{\mathrm{H}} \cdot \boldsymbol{B}_1(n) \mid \boldsymbol{G}(n)], \tag{4.38}$$

and the non-singular matrix $\boldsymbol{L}(n) \in \mathbb{R}^{4 \times 4}$ has been calculated as

$$\boldsymbol{L}(n) \;=\; \left[\begin{array}{c|c} \boldsymbol{0} & \boldsymbol{I}_2 \\ \hline \boldsymbol{I}_2 & \boldsymbol{B}_1(n)^{\mathrm{H}} \boldsymbol{B}_1(n) \end{array}\right]. \tag{4.39}$$

In order to compute the inversion of the formula (4.37), we use again the Woodbury identity according to (3.17) . So the inversion of (4.37)can be calculated as

$$\boldsymbol{W}(n) \;=\; \boldsymbol{W}(n-1) - \boldsymbol{S}(n) \cdot \boldsymbol{\Lambda}(n) \cdot \boldsymbol{S}(n)^{\mathrm{H}}, \tag{4.40}$$

where $\boldsymbol{S}(n)$ is a $d \times 4$ matrix

$$\boldsymbol{S}(n) \;=\; \boldsymbol{W}(n-1) \cdot \boldsymbol{F}(n), \tag{4.41}$$

and $\boldsymbol{\Lambda}(n)$ is a 4×4 matrix,

$$\boldsymbol{\Lambda}(n) \;=\; (\boldsymbol{L}(n)^{-1} + \boldsymbol{S}(n)^{\mathrm{H}} \cdot \boldsymbol{F}(n))^{-1}. \tag{4.42}$$

After a few mathematical derivations, pseudo-inverse of the abstracted signal subspace

$\boldsymbol{E}_1(n)$ can be calculated with respect to the previous moment as

$$\boldsymbol{E}_1(n)^\dagger = \boldsymbol{E}_1(n-1)^\dagger + \boldsymbol{R}(n) \cdot \boldsymbol{T}(n)^{\mathrm{H}}, \tag{4.43}$$

where $\boldsymbol{R}(n)$ is a $d \times 4$ matrix

$$\boldsymbol{R}(n) = \boldsymbol{S}(n) \cdot \boldsymbol{\Lambda}(n), \tag{4.44}$$

and $\boldsymbol{T}(n)$ is a $m \times 4$ matrix,

$$\boldsymbol{T}(n) = \boldsymbol{B}_1(n)[\boldsymbol{I} \mid \boldsymbol{0}] - \boldsymbol{E}_1(n-1) \cdot \boldsymbol{S}(n). \tag{4.45}$$

Now, the update of matrix $\boldsymbol{\Phi}(n)$ can be obtained by substituting each of the equations (4.43) and (4.33) into (4.31) such as

$$\boldsymbol{\Upsilon}(n) = \boldsymbol{\Upsilon}(n-1) + \boldsymbol{Z}(n) \cdot \boldsymbol{Q}(n)^{\mathrm{H}}, \tag{4.46}$$

where $\boldsymbol{Z}(n)$ is a $d \times 6$ matrix

$$\boldsymbol{Z}(n) = [\boldsymbol{E}_1(n-1)^\dagger \cdot \boldsymbol{B}_2(n) \mid \boldsymbol{R}(n)], \tag{4.47}$$

and $\boldsymbol{Q}(n)$ is a $d \times 6$ matrix,

$$\boldsymbol{Q}(n) = [\boldsymbol{G}(n) \mid \boldsymbol{E}_2(n)^{\mathrm{H}} \cdot \boldsymbol{T}(n)], \tag{4.48}$$

The Adaptive Unitary ESPRIT based on FBA-PAST algorithm is summarized in Tab. 4.4.

for the reason that the initial values impact the transient behavior of the algorithm, we chose an initial value only for $\boldsymbol{E}_s(0)$. Then we run the FBA-PAST algrithm to calculate the signal subspace $\boldsymbol{E}_s(1)$ for the first observation as one step to compute the $\boldsymbol{W}(1)$ and $\boldsymbol{\Upsilon}(1)$ insted of giving initial values to them.

1. **FBA-PAST initialization:** (*cf.* Tab. 3.3) $\boldsymbol{E}_s(0)$
2. **FBA-PAST main section:** (*cf.* Tab. 3.3) $\boldsymbol{E}_s(1)$
3. **Unitary-ESPRIT main section:** (*cf.* Tab. 4.3) $\boldsymbol{E}_s(1)^{\dagger}$ and $\boldsymbol{\Upsilon}(1)$
4. **Adaptive Unitary ESPRIT initialization:**
   $\boldsymbol{E}_1(1),\ \boldsymbol{E}_2(1),\ \boldsymbol{W}(1) = [\boldsymbol{E}_1(1)^{\mathrm{H}}\boldsymbol{E}_1(1)]^{-1},\ \boldsymbol{E}_1(1)^{\dagger}$ and $\boldsymbol{\Upsilon}(1)$
   for $n = 2, 3, ...$ do
   - **FBA-PAST main section:** (*cf.* Tab. 3.3)
   - **Unitary ESPRIT main section:**

$$
\begin{aligned}
\boldsymbol{F}(n)^{-1} &= [\boldsymbol{E}_1(n-1)^{\mathrm{H}} \cdot \boldsymbol{B}_1(n) \mid \boldsymbol{G}(n)] \\
\boldsymbol{S}(n) &= \boldsymbol{W}(n-1) \cdot \boldsymbol{F}(n) \\
\boldsymbol{L}(n)^{-1} &= \left[\begin{array}{c|c} -\boldsymbol{B}_1(n)^{\mathrm{H}}\boldsymbol{B}_1(n) & \boldsymbol{I} \\ \hline \boldsymbol{I} & \boldsymbol{0} \end{array}\right] \\
\boldsymbol{\Lambda}(n) &= (\boldsymbol{L}(n)^{-1} + \boldsymbol{S}(n)^{\mathrm{H}} \cdot \boldsymbol{F}(n))^{-1} \\
\boldsymbol{R}(n) &= \boldsymbol{S}(n) \cdot \boldsymbol{\Lambda}(n) \\
\boldsymbol{W}(n) &= \boldsymbol{W}(n-1) - \boldsymbol{S}(n) \cdot \boldsymbol{\Lambda}(n) \cdot \boldsymbol{S}(n)^{\mathrm{H}} \\
\boldsymbol{T}(n) &= \boldsymbol{B}_1(n)[\boldsymbol{I} \mid \boldsymbol{0}] - \boldsymbol{E}_1(n-1) \cdot \boldsymbol{S}(n) \\
\boldsymbol{Z}(n) &= [\boldsymbol{E}_1(n-1)^{\dagger} \cdot \boldsymbol{B}_2(n) \mid \boldsymbol{R}(n)] \\
\boldsymbol{Q}(n) &= [\boldsymbol{G}(n) \mid \boldsymbol{E}_1(n)^{\mathrm{H}} \cdot \boldsymbol{T}(n)] \\
\boldsymbol{E}_1(n)^{\dagger} &= \boldsymbol{E}_1(n-1)^{\dagger} + \boldsymbol{R}(n) \cdot \boldsymbol{T}(n)^{\mathrm{H}} \\
\boldsymbol{\Upsilon}(n) &= \boldsymbol{\Upsilon}(n-1) + \boldsymbol{Z}(n) \cdot \boldsymbol{Q}(n)^{\mathrm{H}}
\end{aligned}
$$

   - $\{\mu_i\}_{1 \le i \le d} = 2\arctan(\mathrm{eig}\{\boldsymbol{\Upsilon}(n)\})$

Tab. 4.4: Summary of Adaptive Unitary ESPRIT based on FBA-PAST algorithm

4.5 Adaptive 2-D Unitary ESPRIT algorithm based on the FBA-PAST subspace tracker

We extend the Adaptive Unitary ESPRIT algorithm based on the FBA-PAST subspace tracker to 2-D as shown in Tab. 4.5.

1. **FBA-PAST initialization:** (*cf.* Tab. 3.3) $\boldsymbol{E}_s(0)$
2. **FBA-PAST main section:** (*cf.* Tab. 3.3) $\boldsymbol{E}_s(1)$
3. **2-D Unitary ESPRIT:** (*see* [HZMN95]) $\boldsymbol{E}^{(r)}(1)^{\dagger}$ and $\boldsymbol{\Upsilon}^{(r)}(1),\ r = 1, 2$
4. **Adaptive Unitary ESPRIT initialization:**
   $\boldsymbol{E}_1^{(r)}(1),\ \boldsymbol{E}_2^{(r)}(1),\ \boldsymbol{W}^{(r)}(1) = [\boldsymbol{E}_1^{(r)}(1)^{\mathrm{H}}\boldsymbol{E}_1^{(r)}(1)]^{-1},\ \boldsymbol{E}_1^{(r)}(1)^{\dagger}$ and $\boldsymbol{\Upsilon}^{(r)}(1)$
   for n = 2,3,...
   for r = 1,2
   - **FBA-PAST main section:** (*cf.* Tab. 3.3)
   - **Adaptive Unitary ESPRIT main section:**(*cf.* Tab. 4.4)
   - $\{\mu_i^{(r)}\}_{1\leq i\leq d} = 2\arctan(\mathrm{eig}\{\boldsymbol{\Upsilon}^{(r)}(n)\})$

Tab. 4.5: Summary of Adaptive 2-D Unitary ESPRIT based on FBA-PAST algorithm

4.6 Simulation Results

Similarly to the previous chapter's scenarios, we are going to track the DOA of three sources for two scenarios. In the first scenario, we consider two of sources are slowly moving overall snapshots and the third one has a constant DOA, as shown in Fig. 3.4. In the second scenario, we consider two of sources are crossing at the half of distance and the third has a constant DOA, as shown in Fig. 3.11. And the URA has 6×6 sensors, and the forgetting factor has been chosen to be 0.97 at length window $N = 1000$, according to [Yan95]. In each scenario, we evaluate the Adaptive 2-D Unitary ESPRIT algorithm compare to the Adaptive ESPRIT based on PAST algorithm and the 2-D Unitary ESPRIT based on FBA-PAST algorithm for three cases:

- SNR = 0 dB and $\rho = 0$,
- SNR = 0 dB, $\rho = 0.99$, and $\rho = 0.5$ for the first and second scenarios respectively, and
- SNR = -3 dB and $\rho = 0$, for both scenarios, respectively.

Here, the initial values $\boldsymbol{P}(0)$, $\boldsymbol{W}(0)$, $\boldsymbol{\phi}(0)$, and $\boldsymbol{P}^{\mathrm{FBA}}(0)$ have been chosen to be $d \times d$ identity matrix, and the signal subspaces $\boldsymbol{U}_s$, and $\boldsymbol{E}_s$ have been included d orthonormal vectors. We should note that, the initial values only impact the transient behavior but the performance of the algorithm remains stable.

The simulation results are depicted for the first scenario as follows: Fig. 4.4 and Fig. 4.5,(a) and (b) visualize the first case, Fig. 4.6 and Fig. 4.7,(a) and (b) depict the second case, and Fig. 4.8 and Fig. 4.9,(a) and (b) present the third case, respectively. Similarly to second scenario in the previous chapter, Fig. 4.10 and Fig. 4.11,(a) and (b), Fig. 4.12 and Fig. 4.13,(a) and (b), and Fig. 4.14 and Fig. 4.15,(a) and (b) visualize the first case, second and third case, respectively.

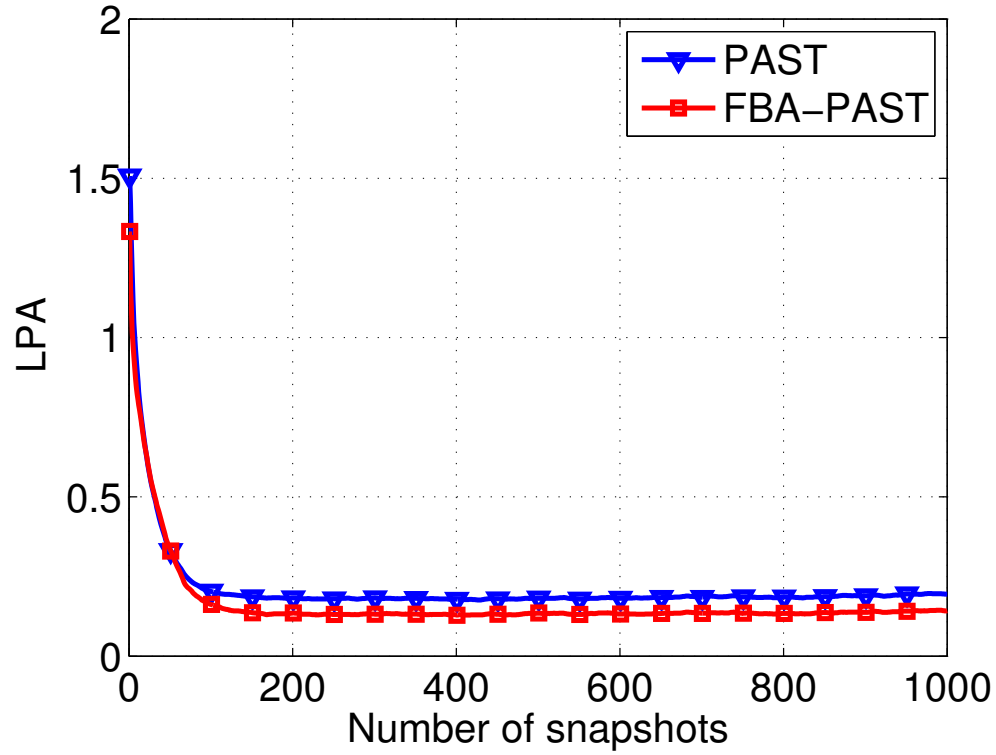
For the first case, when the SNR equals 0 dB and no correlation between the sources for scenarios 1 and 2, it can been found that performance of Adaptive Unitary ESPRIT based on FBA-PAST algorithm outperforms the performance of Adaptive ESPRIT based on PAST algorithm due to use the FBA processing. Moreover, the performance of Adaptive ESPEIT and Adaptive Unitary ESPRIT was similar to them counterpart by using ESPRIT and Unitary ESPRIT based on the PAST and FBA-PAST algorithms, respectively. Moreover, all algorithms based on the FBA processing require lower calculations than similar algorithms based on standard ESPRIT. We should mentioned also that Adaptive Unitary ESPRIT needs lower computations than Unitary ESPRIT due to updating the invariance of the signal subspaces in addition to the tracking the phase delays matrix.

For the second case, when consider the case of SNR = 0 dB and the correlation between the sources has been increased to be 0.99 for scenario 1; and $\rho = 0.5$ for scenario 2. It can be seen that the performance of each ESPRIT and Adaptive ESPRIT gradate compared to Adaptive Unitary ESPRIT and 2-D unitary ESPRIT. More precisely insight in the first scenario can be shown in Fig. 4.7,(a) and (b). Here, it can be found that the Adaptive ESPRIT was able to track only one sources and failed in tracking the others for the first scenario. For the second scenario as visualized in Fig. 4.13,(a) and (b), the tracking performance failed for all algorithms at the high correlations between the sources and the spatial frequencies as well, i.e., when the correlations between the spatial frequencies and the sources increase the performance degrades.

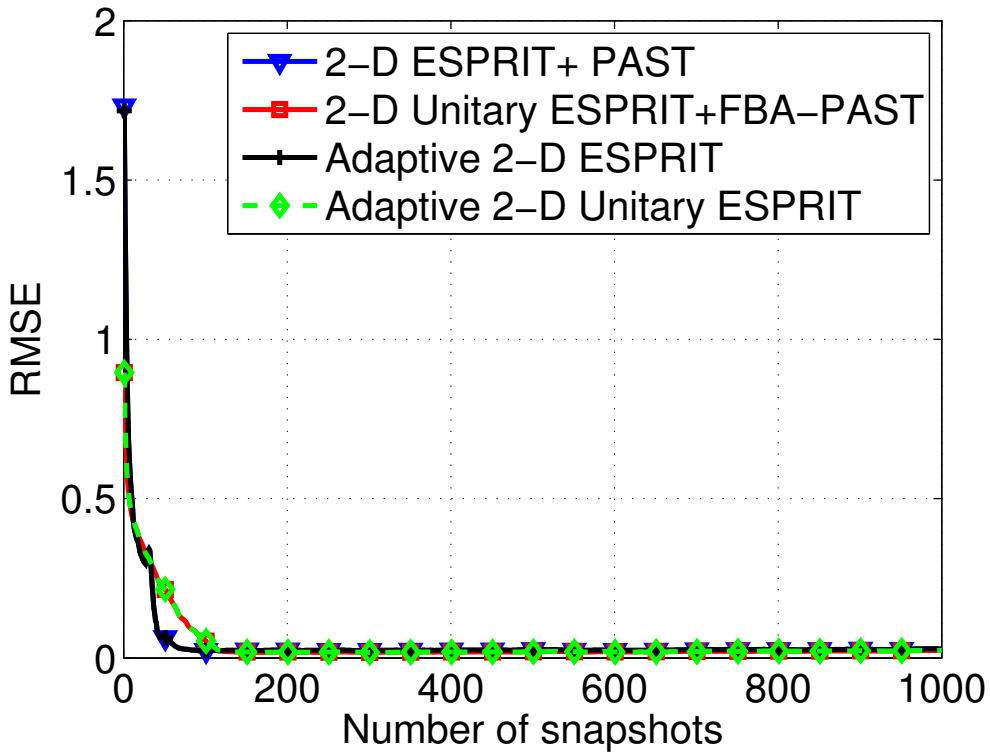
For the third case at SNR = -3 dB and there is no correlation between the sources, it can be observed that performance of Adaptive ESPRIT based on PAST algorithm is lower robustness than Adaptive Unitary ESPRIT based on FBA-PAST. In other words, Adaptive ESPRIT required duple number of observation to perform the same performance like Adaptive Unitary ESPRIT, specially at the beginning of observation as shown in Fig. 4.9,(a) and (b) and Fig. 4.15,(a) and (b), respectively.

Here, it can be seen that the real-valued signal subspaces tracking are higher accurate than its similar by using PAST algorithm. However, each ESPRIT and Unitary ESPRIT as well as Adaptive ESPRIT and Adaptive Unitary ESPRIT were able to track the DOA correctly after a few number of observations. In addition, at the high

correlation between the spatial frequencies, better performance can be observed for the algorithms that based on incorporating the FBA processing.

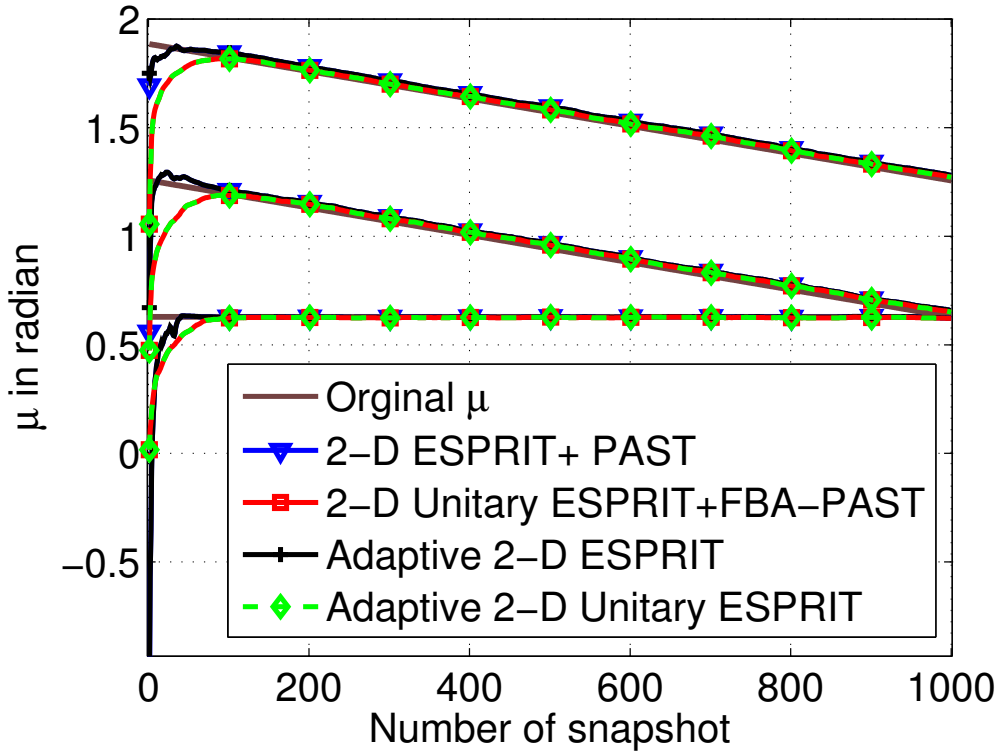


(a) LPA vs. the number of snapshots

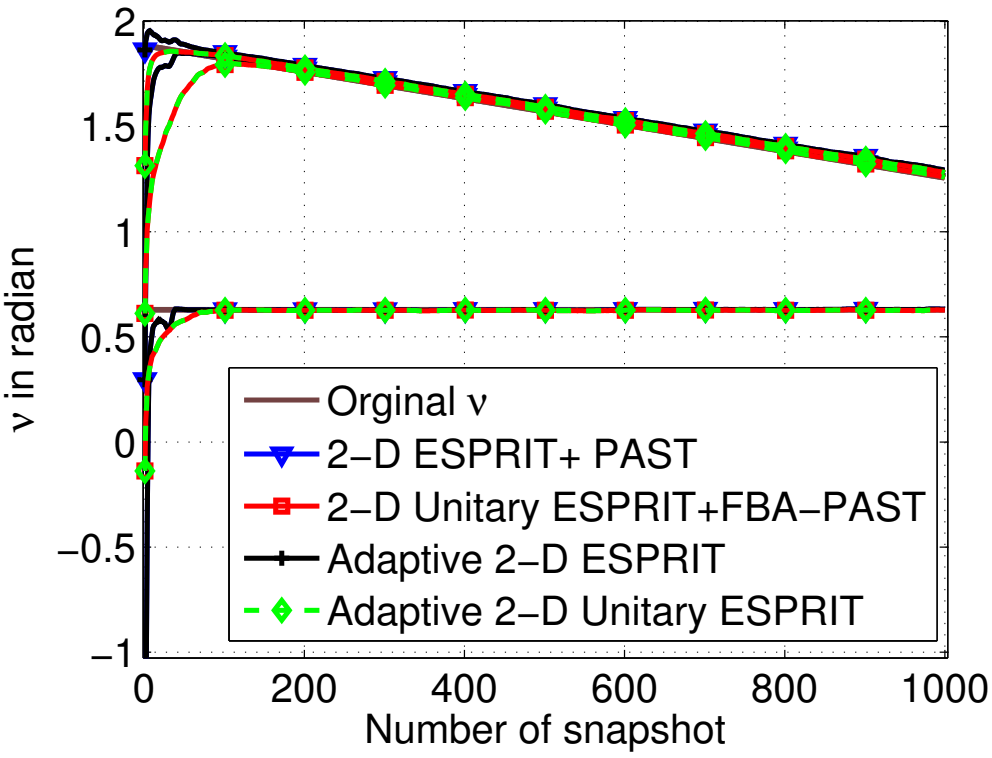


(b) RMSE vs. the number of snapshots

Fig. 4.4: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0$ for scenario 1

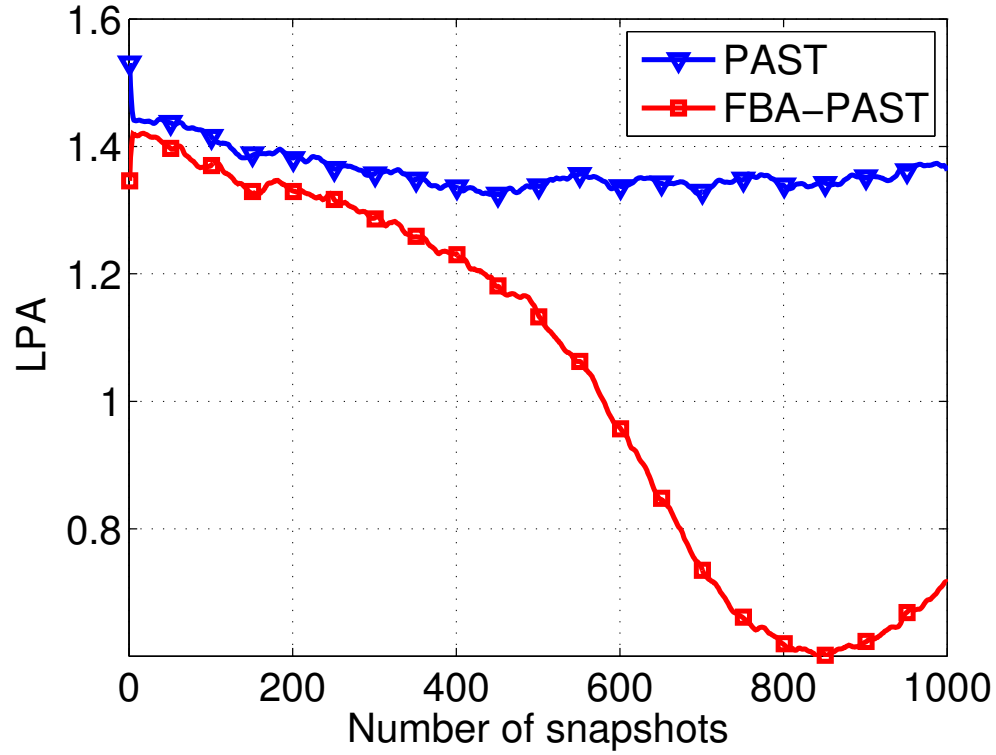


(a) Estimated spatial frequencies on the horizontal plane

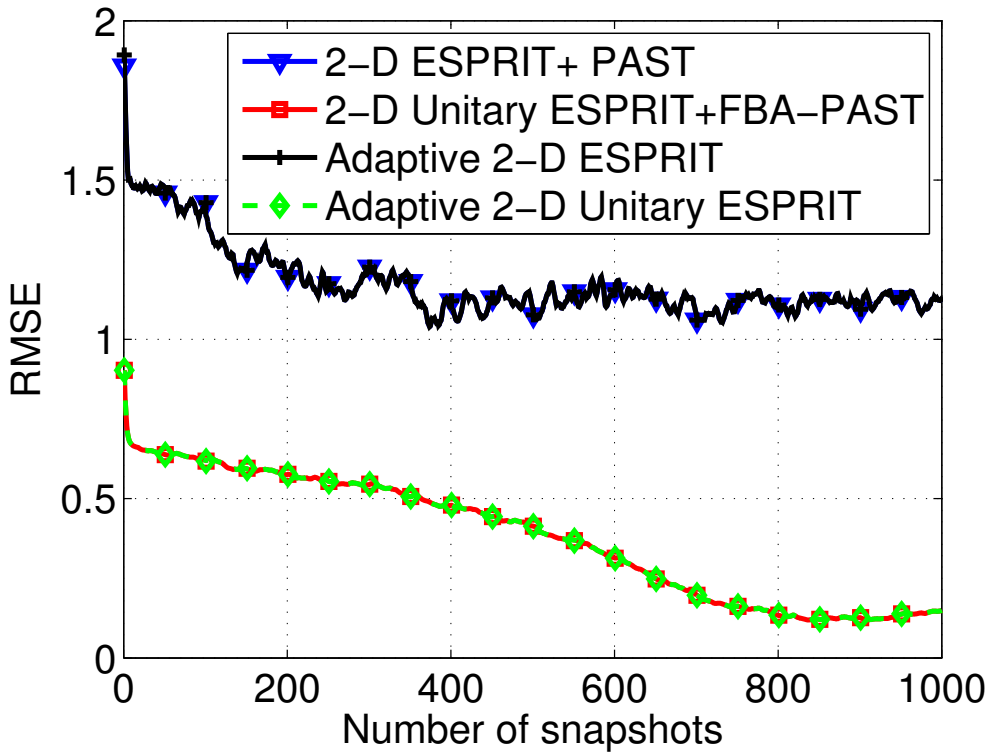


(b) Estimated spatial frequencies on the vertical plane

Fig. 4.5: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0$ for scenario 1

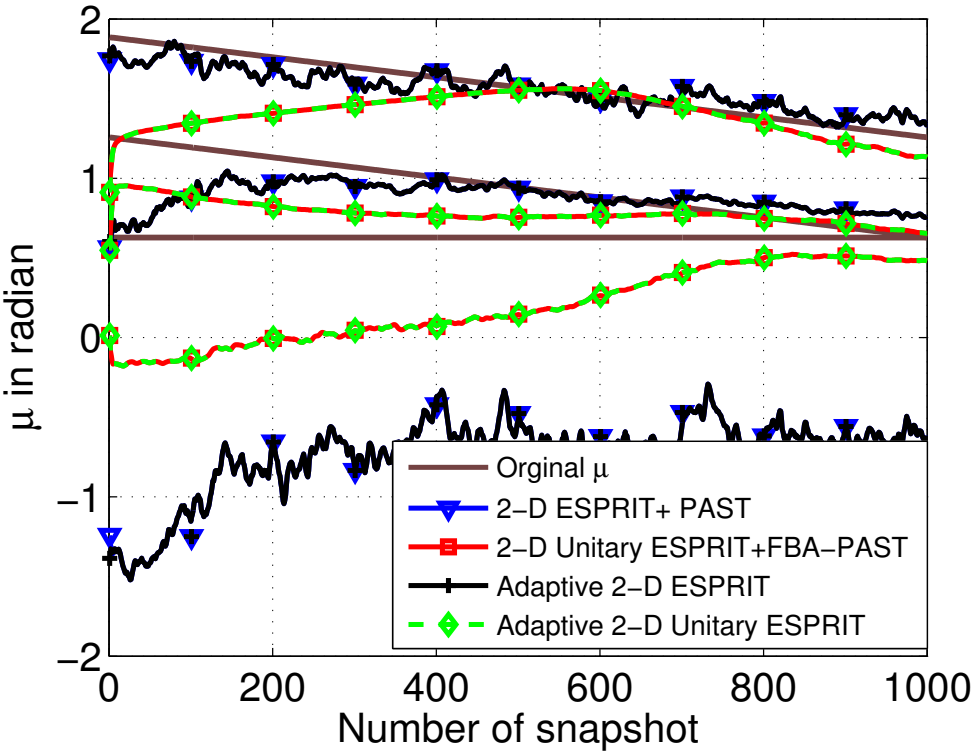


(a) LPA vs. the number of snapshots

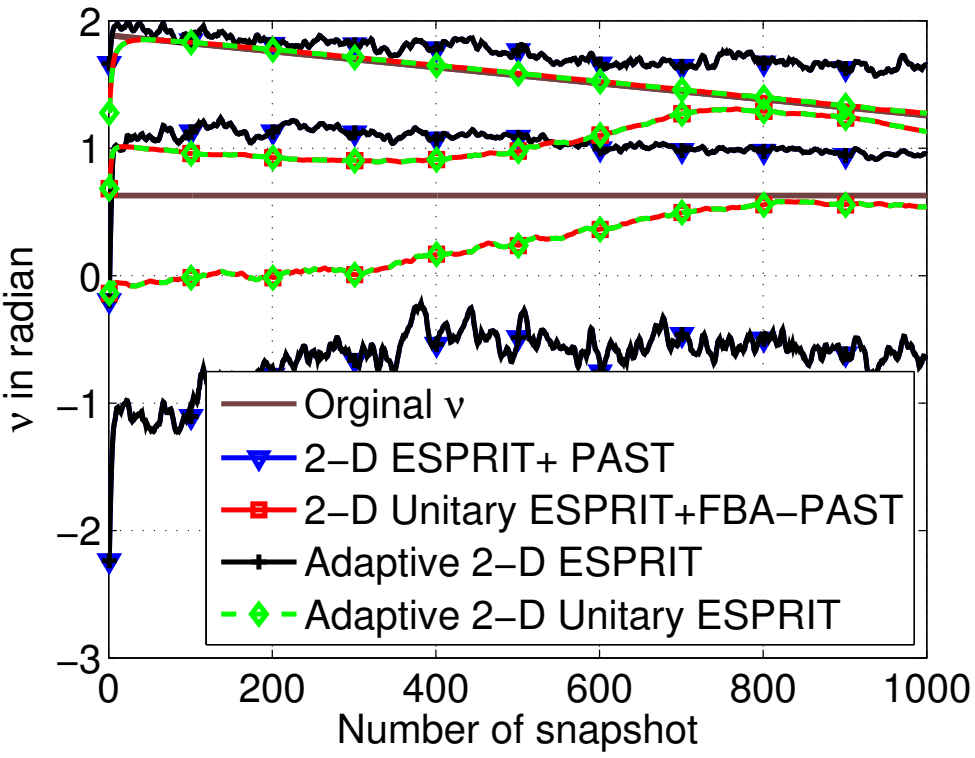


(b) RMSE vs. the number of snapshots

Fig. 4.6: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0.99$ for scenario 1

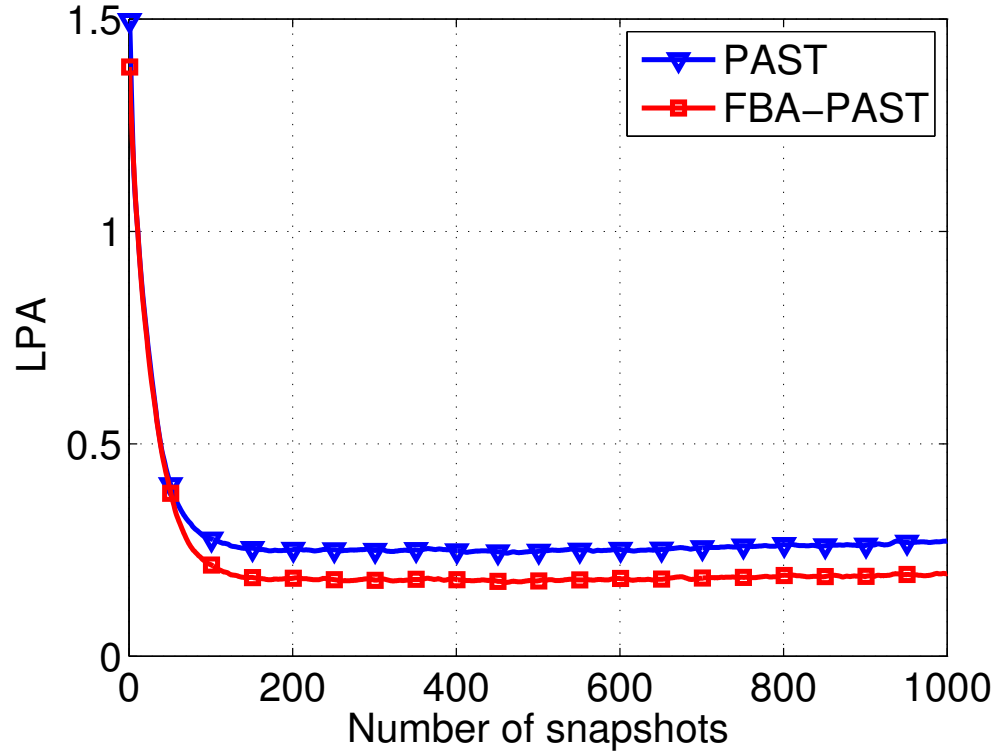


(a) Estimated spatial frequencies on the horizontal plane

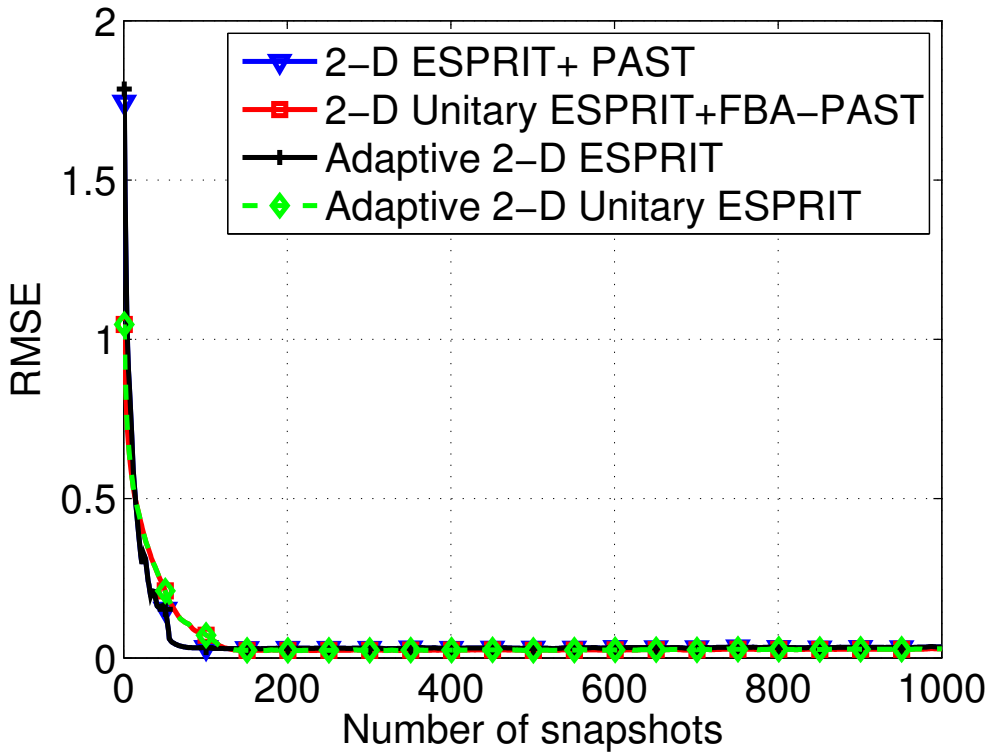


(b) Estimated spatial frequencies on the vertical plane

Fig. 4.7: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0.99$ for scenario 1

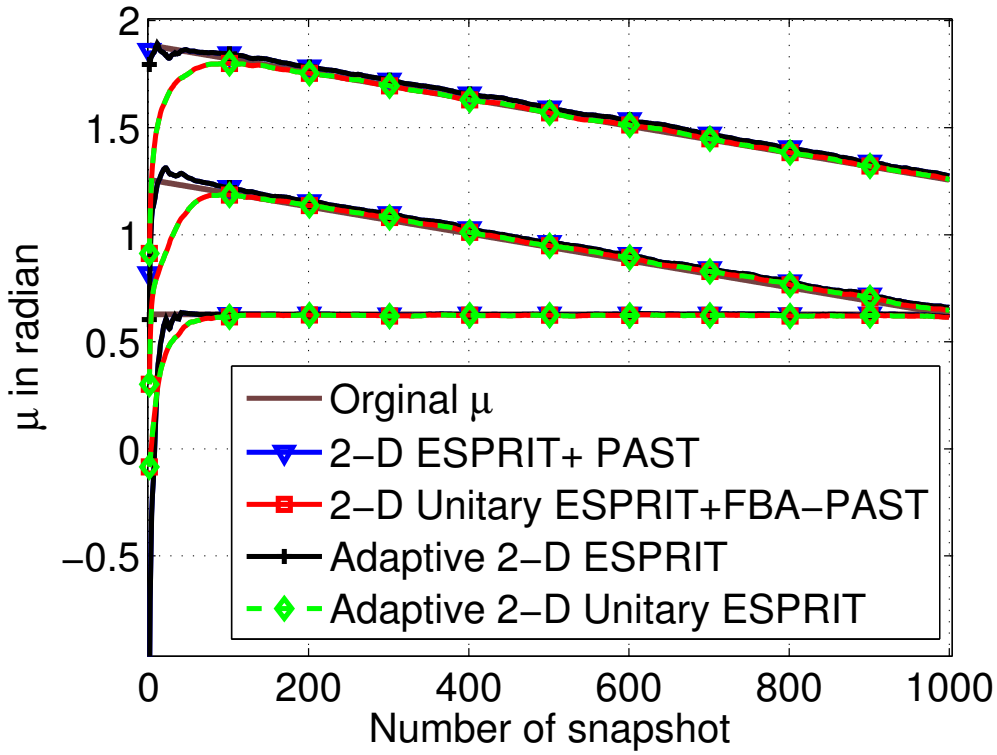


(a) LPA vs. the number of snapshots

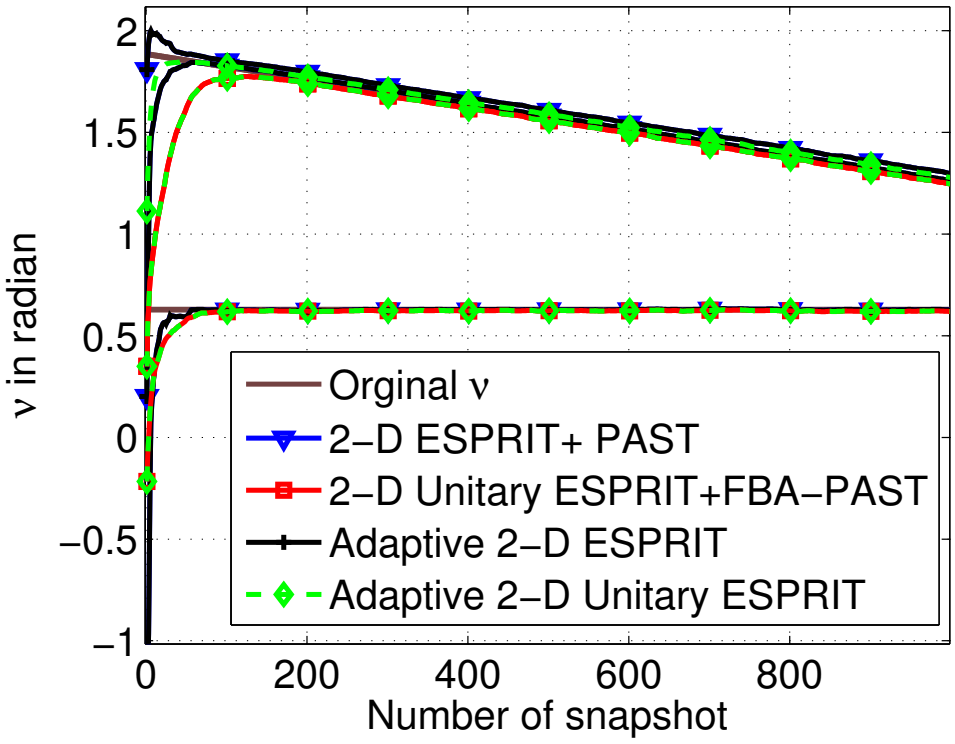


(b) RMSE vs. the number of snapshots

Fig. 4.8: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= -3$ dB and $\rho = 0$ for scenario 1

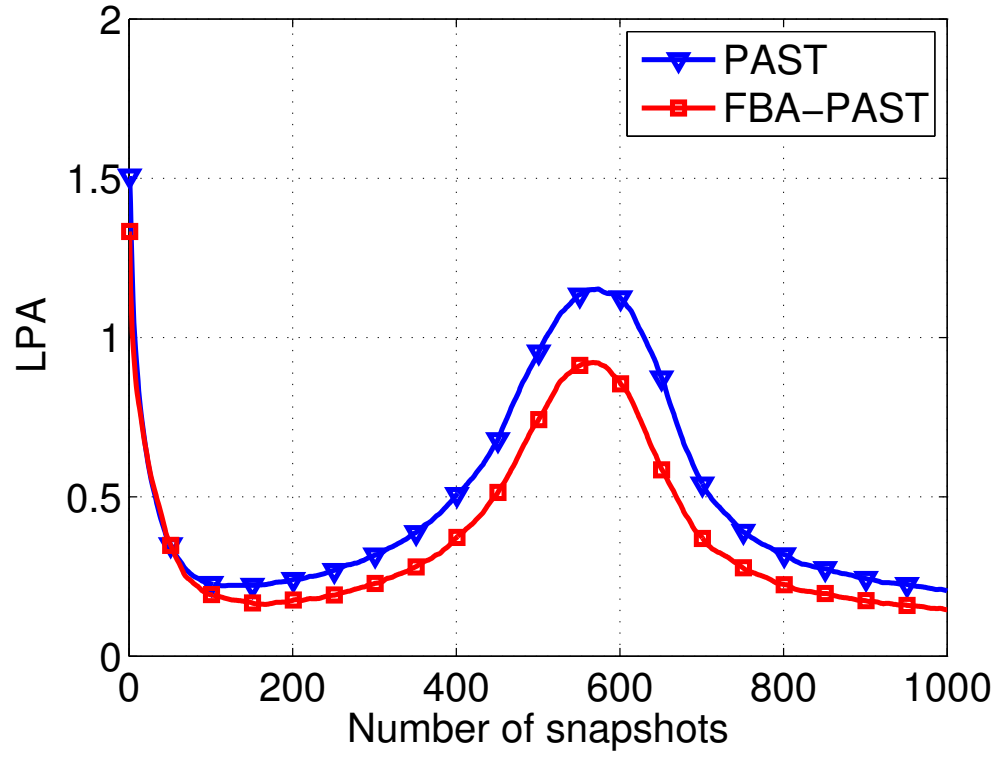


(a) Estimated spatial frequencies on the horizontal plane

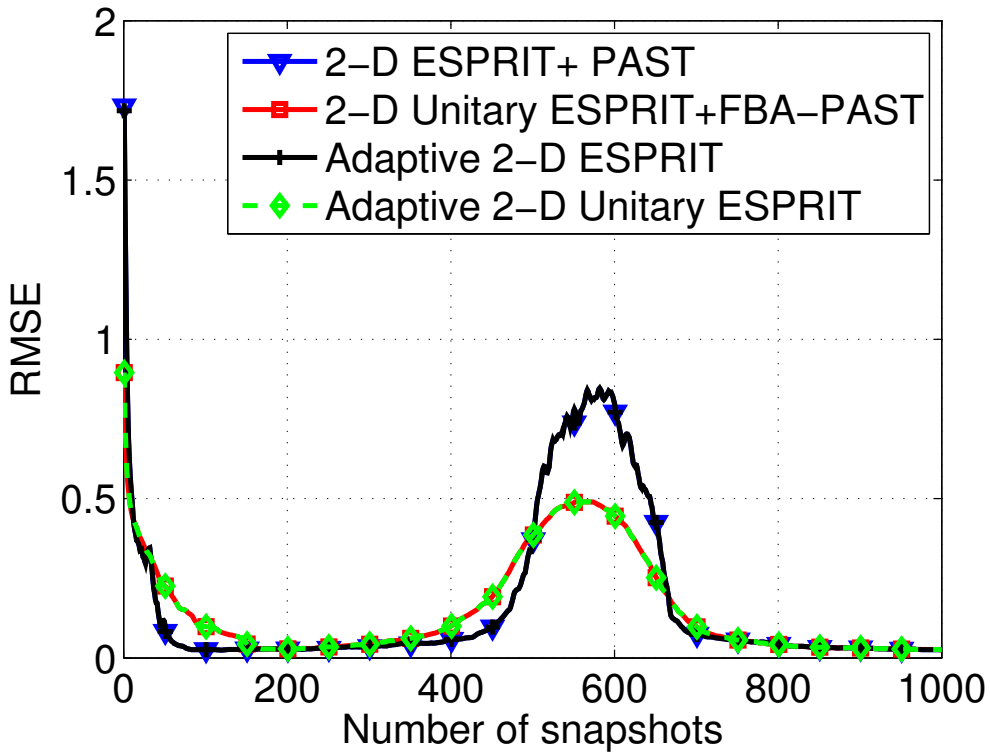


(b) Estimated spatial frequencies on the vertical plane

Fig. 4.9: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= -3$ dB and $\rho = 0$ for scenario 1

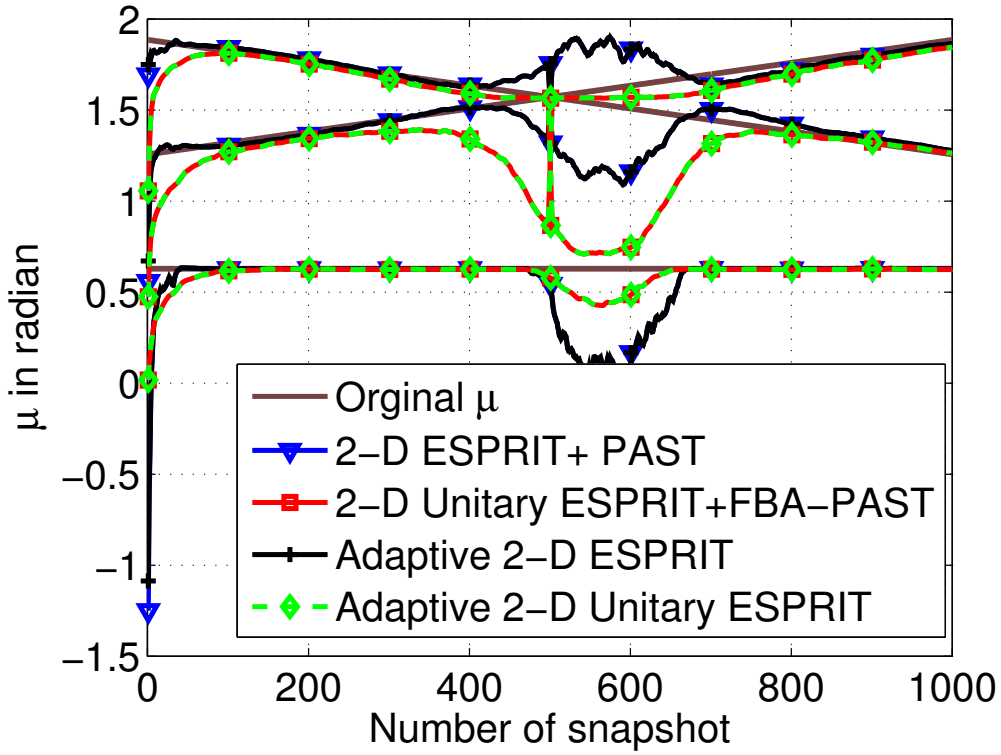


(a) LPA vs. the number of snapshots

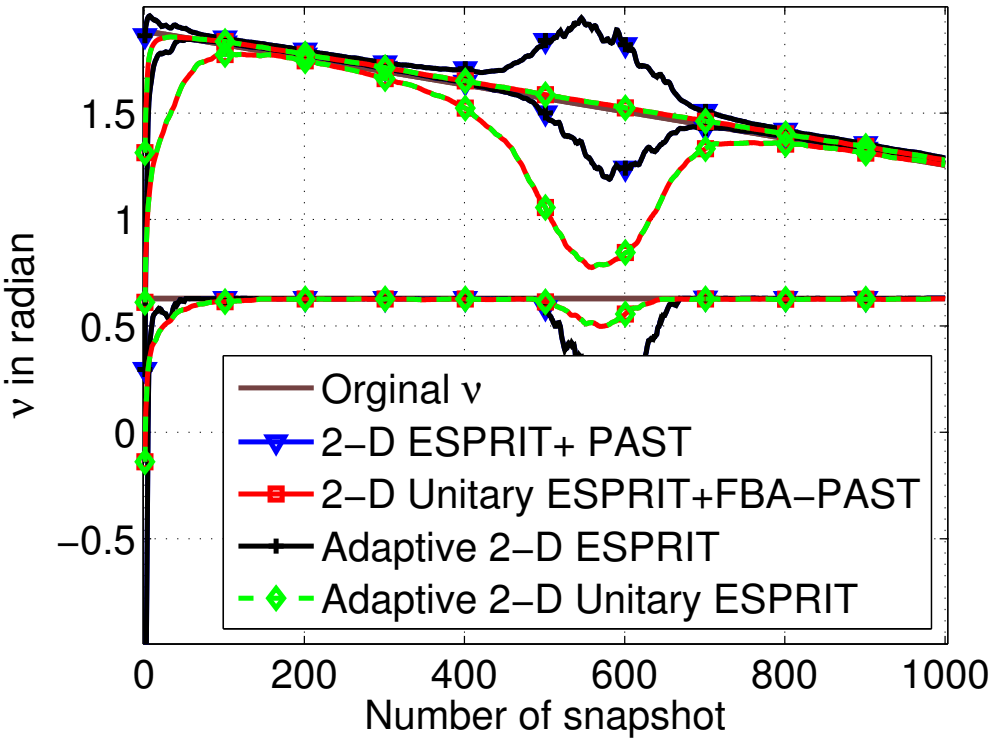


(b) RMSE vs. the number of snapshots

Fig. 4.10: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0$ for scenario 2

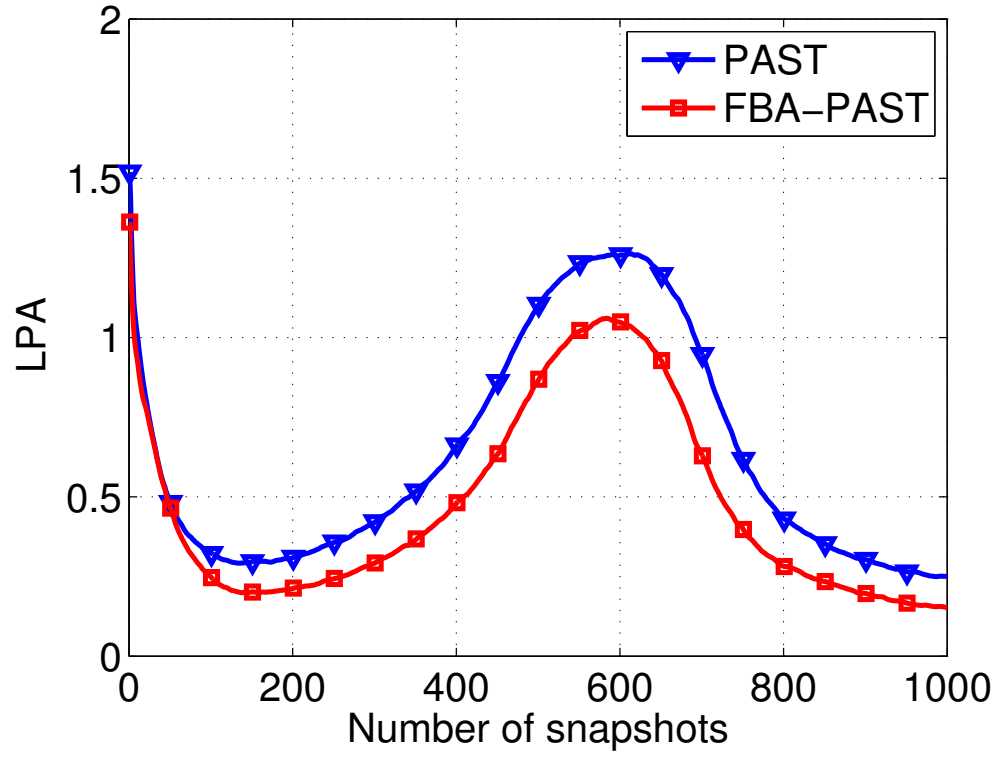


(a) Estimated spatial frequencies on the horizontal plane

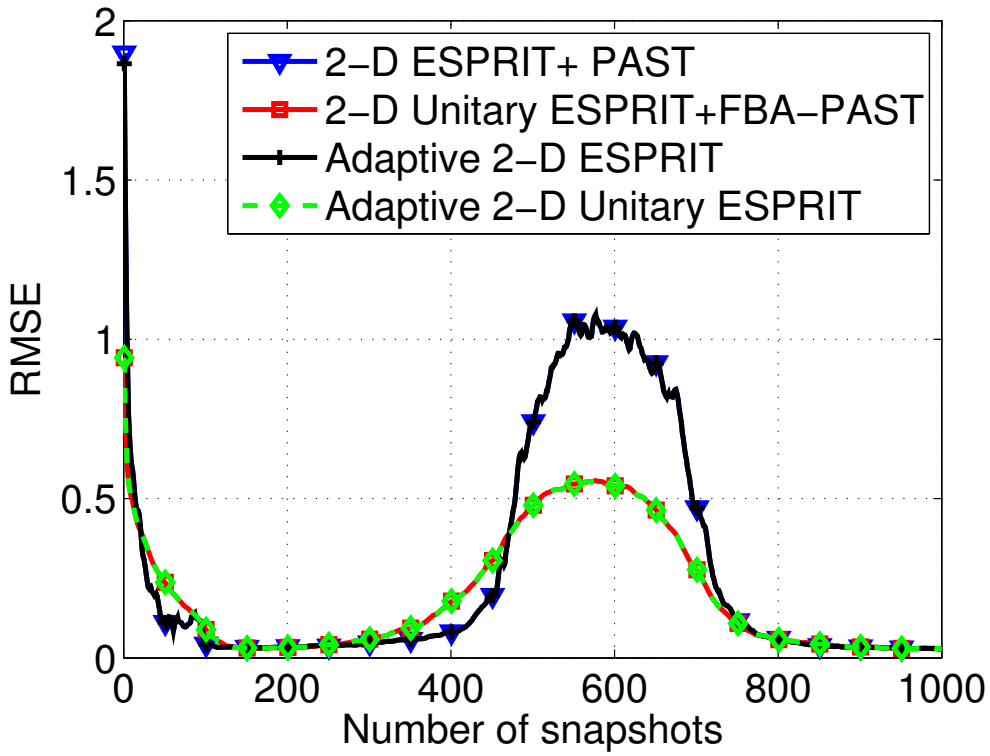


(b) Estimated spatial frequencies on the vertical plane

Fig. 4.11: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0$ for scenario 2

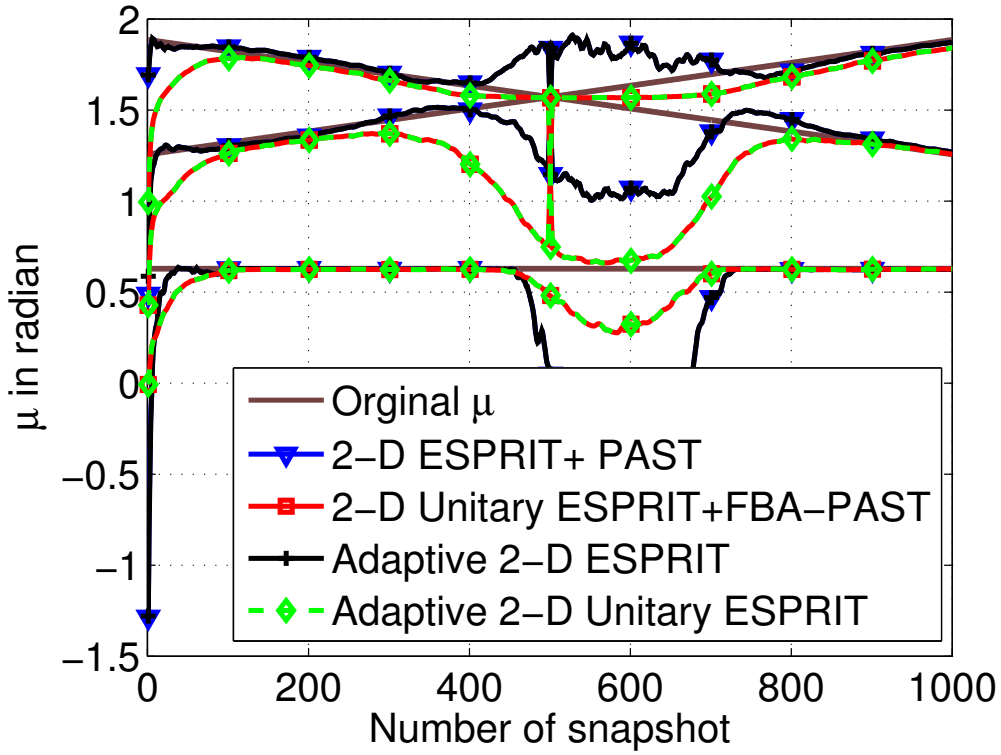


(a) LPA vs. the number of snapshots

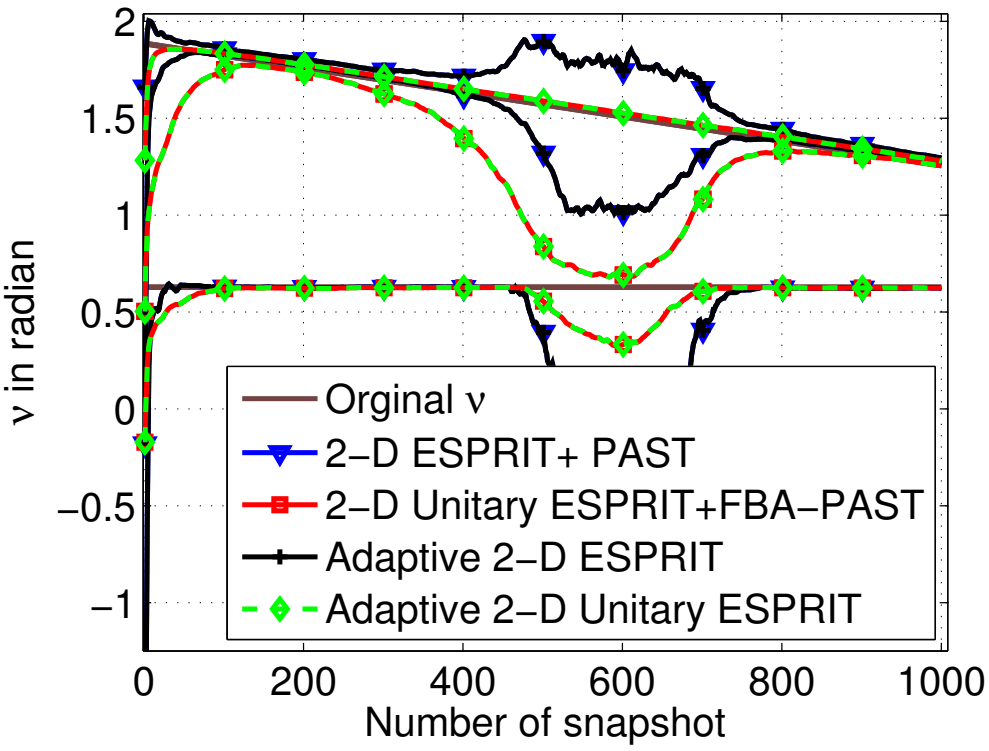


(b) RMSE vs. the number of snapshots

Fig. 4.12: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0.5$ for scenario 2

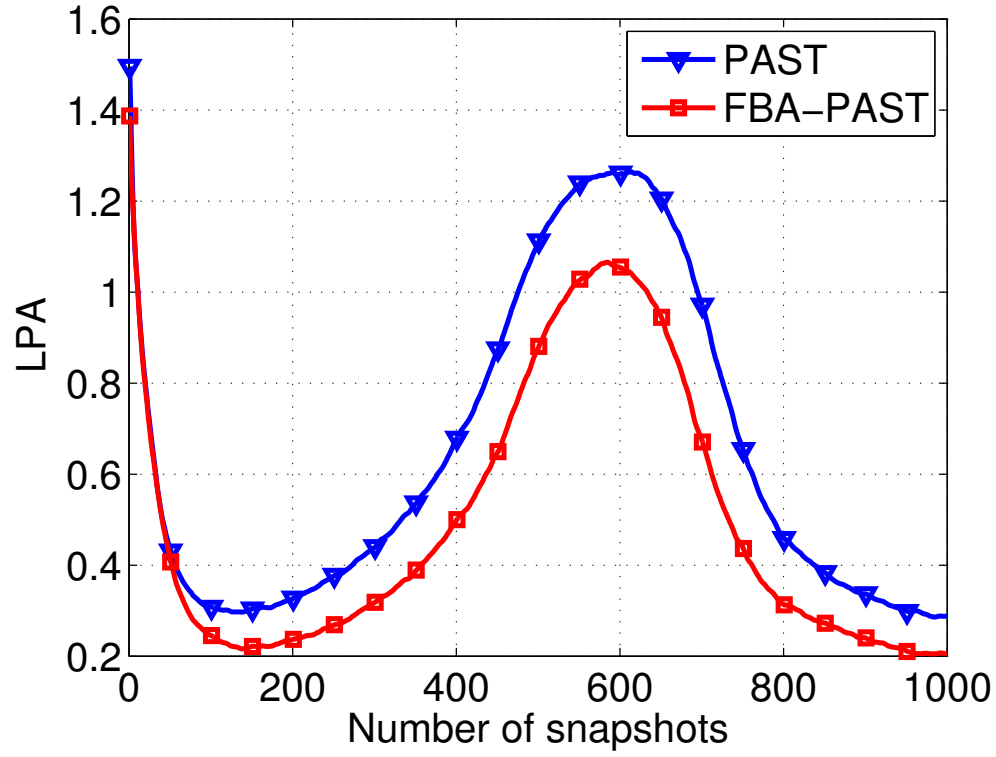


(a) Estimated spatial frequencies on the horizontal plane

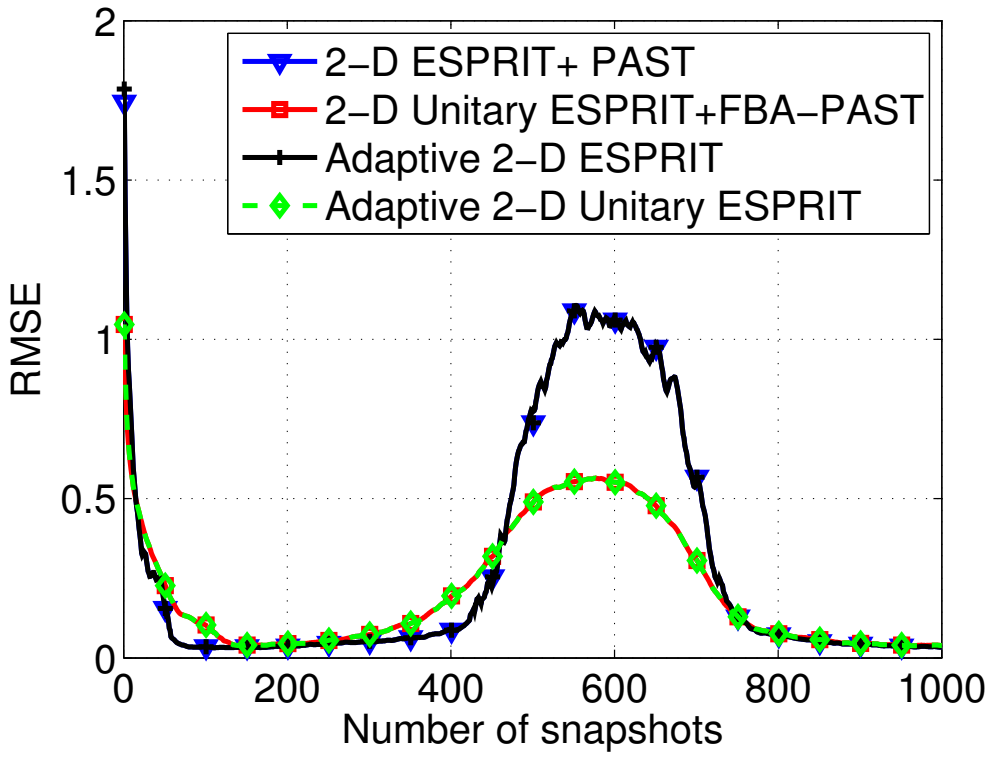


(b) Estimated spatial frequencies on the vertical plane

Fig. 4.13: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= 0$ dB and $\rho = 0.5$ for scenario 2

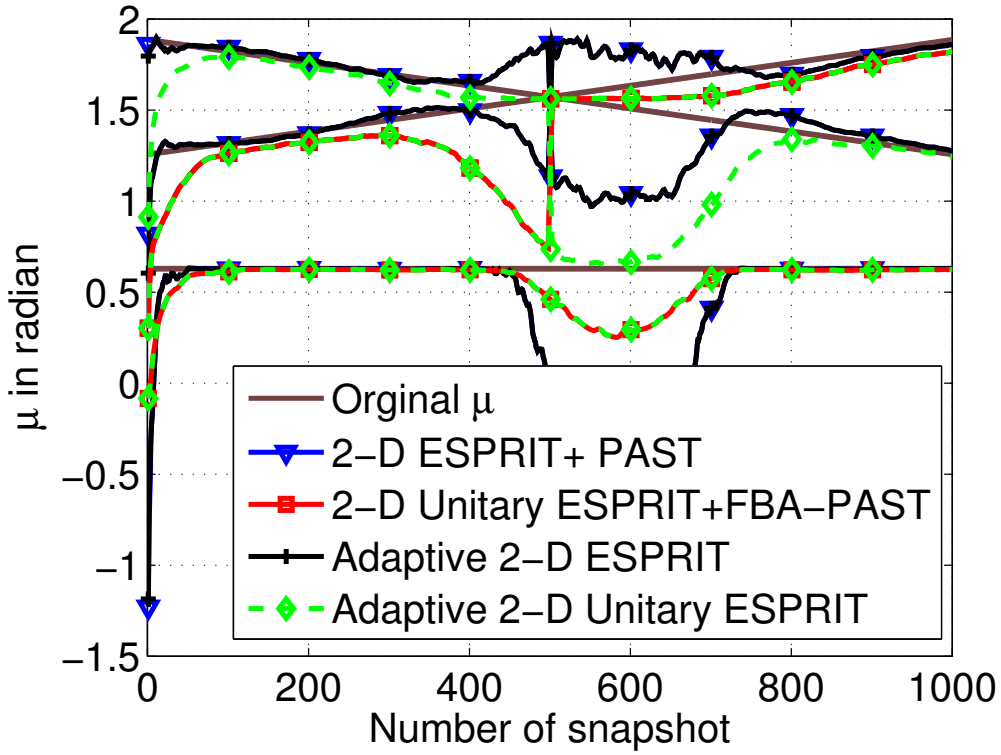


(a) LPA vs. the number of snapshots

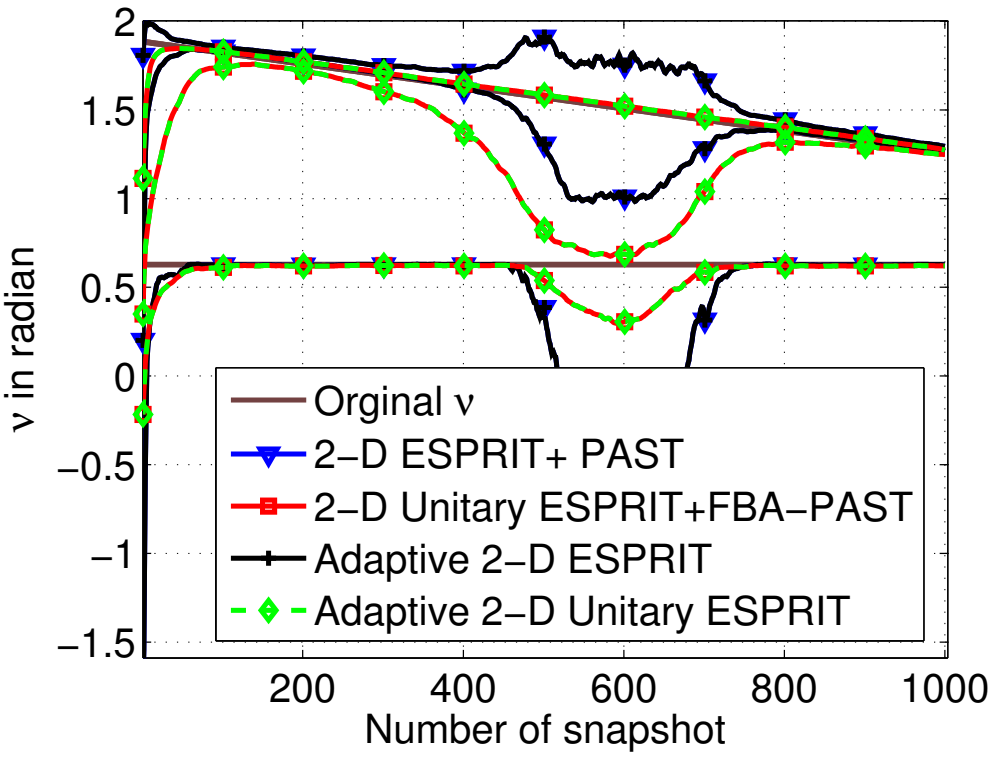


(b) RMSE vs. the number of snapshots

Fig. 4.14: LPA and RMSE vs. the number of snapshots for 3 sources impinging on a URA of 6×6 sensors at SNR $= -3$ dB and $\rho = 0$ for scenario 2



(a) Estimated spatial frequencies on the horizontal plane



(b) Estimated spatial frequencies on the vertical plane

Fig. 4.15: Estimated spatial frequencies for 3 sources impinging on a URA of 6×6 sensors at SNR $= -3$ dB and $\rho = 0$ for scenario 2

5. CONCLUSIONS AND FUTURE WORK

Our aim in this thesis was to improve the tensor-based subspace tracking scheme by incorporating the FBA processing and applying it consequently to track the signal parameters, i.e., tracking the real-valued subspaces based on incorporating the FBA processing, then estimating the DOA in non-stationary scenarios.

We showed that by including FBA the accuracy of the tensor-based subspace tracking algorithm was improved and it outperformed its matrix-based counterpart. In addition the proposed subspace tracking algorithm is robust for critical scenarios and has a lower mathematical complexity compared to some existing algorithms.

We summarized in the second chapter of the thesis the tensor algebra tools that needed to be used throughout the thesis according to [dLdMV00]. Then we studied one application of using the HOSVD in estimating the subspace for multidimensional harmonic retrieval problems recording to [HRDG08].

In the third chapter of the thesis, we improved Extended PAST [CH13] that was required for a generic framework for tensor subspace tracking via Kronecker structured projections. Here, we modified the PAST algorithm to track the real-valued subspaces by incorporating the FBA processing. Therefore, the mathematical complexity has been reduced and at the same time the tracking accuracy became better. Similar to [CH13], we developed an extended version of the PAST algorithm again for tracking the real-valued n-mode unfolding subspaces. As we show, both of the proposed subspace tracking algorithms, FBA-PAST and Extended FBA-PAST, provide a better performance in terms of Large Principal Angle (LPA), e.g, the accuracy in tracking the signal subspaces becomes better. Moreover, the mathematical computations have been performed in the real domain instead of the complex domain. In addition, further robustness in signal parameters tracking has been obtained at critical scenarios. This has been observed as well in the Root Mean Square Error (RMSE) criteria after using some DOA estimation algorithms like 2-D ESPRIT, 2-D Unitary ESPRIT, STE and UTE. Furthermore, 2-D Unitary ESPRIT and UTE based on FBA-PAST and Extended FBA-PAST outperformed their standard versions 2-D ESPRIT and STE. On the other hand, the tensor-based techniques, STE and UTE, achieved a better performance compared to their matrix-based counterparts.

In the fourth chapter of the thesis, we reviewed ESPRIT and Unitary ESPRIT as well as Adaptive ESPRIT based on the PAST tracking algorithm. We developed an adaptive version of Unitary ESRPIT based on the FBA-PAST algorithm. we found that, the Adaptive Unitary ESPRIT algorithm is be able to track the signal parameters based on the FBA-PAST algorithm with the same performance but lower mathematical complexity and better performance compared to Adaptive ESPRIT algorithm. Moreover, we extended the Adaptive Unitary ESPRIT based on FBA-PAST from 1-dimension to 2-dimension.

As we can see, the mathematical complexity is still an open research issue in tensor-based subspaces tracking as well as matrix-based subspace tracking. In other words, it is of interest to further reduce the complexity of the tensor-based subspace tracking via Kronekcer structured projections.

More improvement can be found in tracking the signal subspaces in [BDR05] and [BRD05] for the fast approximated power iteration subspace tracking, and the fast Adaptive ESPRIT algorithm, respectively. Moreover, the performance of the tracking algorithm is related as well to the speed of signal subspaces changing. As shown in the PAST algorithm that its performance will degrade considerably when the subspace changes fast. A new robust Kalman Filter based subspace tracking algorithm can be proposed also to improve the performance of tracking time-variant subspaces according to [LZC10]. Furthermore, incorporating FBA can be performed with the aforementioned algorithms to upgrade the performance of some algorithms as well as to reduce the mathematical complexity. Last but not the least, Structured Least Squares (SLS) [HN96] can be applied to improve the performance of Adaptive ESPRIT as well as Adaptive Unitary ESPRIT.

BIBLIOGRAPHY

[BDR05] R. Badeau, B. David, and G. Richard. Fast approximated power iteration subspace tracking. *IEEE Transactions on Signal Processing*, 53(8):2931–2941, 2005.

[BRD03] R. Badeau, G. Richard, and B. David. Adaptive esprit algorithm based on the past subspace tracker. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '03)*, volume 6, 2003.

[BRD05] R. Badeau, G. Richard, and B. David. Fast adaptive esprit algorithm. In *Statistical Signal Processing, 2005 IEEE/SP 13th Workshop on*, pages 289–294, 2005.

[CH13] F. Roemer E.-K. Kasnakli Y. Cheng and M. Haardt. Tensor subspace tracking via kronecker structured projections (TeTraKron). *IEEE*, 5th Int., Dec. 2013.

[dLdMV00] L. de Lathauwer, B. de Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21, No. 4:12531278, 2000.

[EY36] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[GL96] G. H. Golub and C. F. Van Loan. *Matrix computations*. the Johns Hopkins University Press, 3 edition, 1996.

[HN95] M. Haardt and J.A. Nossek. Unitary esprit: how to obtain increased estimation accuracy with a reduced computational burden. *Signal Processing, IEEE Transactions on*, 43(5):1232–1242, 1995.

[HN96] M. Haardt and J.A. Nossek. Structured least squares to improve the performance of esprit-type high-resolution techniques. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 5, pages 2805–2808 vol. 5, 1996.

[HRDG08] M. Haardt, F. Roemer, and G. Del Galdo. Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems. *IEEE Transactions on Signal Processing*, 56(7):3198–3213, July 2008.

[HZMN95] M. Haardt, M.D. Zoltowski, Cherian P. Mathews, and J.A. Nossek. 2d unitary esprit for efficient 2d parameter estimation. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 3, pages 2096–2099 vol.3, 1995.

[KB09] T. G. Kolda and B. W. Bader. Tensor decomposition and applications. *SIAM*, Review 51(3):455–500, 2009.

[LZC10] B. Liao, Z. G. Zhang, and S. C. Chan. A new robust kalman filter-based subspace tracking algorithm in an impulsive noise environment. *IEEE˙J˙CASII˙EB*, 57(9):740–744, 2010.

[PP12] K. B. Petersen and M. S. Pedersen. The matrix cookbook, nov 2012. Version 20121115.

[RBHW09a] F. Roemer, H. Becker, M. Haardt, and M. Weis. Analytical performance evaluation for hosvd-based parameter estimation schemes. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on*, pages 77–80, 2009.

[RBHW09b] F. Roemer, H. Becker, M. Haardt, and M. Weis. Analytical performance evaluation for HOSVD-based parameter estimation schemes. In *Proc. 3rd IEEE Int Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) Workshop*, pages 77–80, 2009.

[RPK86] R. Roy, A. Paulraj, and T. Kailath. Estimation of signal parameters via rotational invariance techniques - ESPRIT. In *Proc. IEEE Military Communications Conf. - Communications-Computers: Teamed for the 90's MILCOM 1986*, volume 3, 1986.

[Sch86] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Trans. on Antennas and Propagation*, 34(3):276–280, 1986.

[SK93] A.L. Swindlehurst and T. Kailath. Azimuth/elevation direction finding using regular array geometries. *Aerospace and Electronic Systems, IEEE Transactions on*, 29(1):145–156, 1993.

[Str93] G. Strang. The fundamental theorem of linear algebra. *The American Mathematical Monthly*, 100:848–855, Nov., 1993.

[Tuc66] LedyardR Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[Yan95] B. Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43(1):95–107, 1995.

THESES

T1: For multidimensional data, where a multidimensional structure is inherent in the data, Higher-Order SVD (HOSVD) improved estimates of the subspace compared to the SVD concept.

T2: If the number of observations is small or the sources are highly correlated, incorporating Forward Backward Averaging (FBA) can further improve the performance of tracking.

T3: Calculating the core tensor is not required for structured projections.

T4: For time-varying multidimensional data, Extended PAST based on the tensor-based subspace tracking via Kronecker structured projections (TeTraKron) framework outperforms the matrix-based approach PAST.

T5: Including FBA involves only real-valued processing.

T6: Incorporating FBA leads to an improved accuracy of the subspace tracking and a lower computational complexity.

T7: Extended FBA-PAST algorithms outperforms the original PAST and the Extended-PAST algorithms in terms of robustness, accuracy, and involves only real-valued processing.

T8: 2-D Unitary ESPRIT based on FBA-PAST gives better performance compared to 2-D ESPRIT based on PAST.

T9: UTE based on Extended FBA-PAST outperforms STE based on Extended PAST.

T10: Adaptive Unitary ESPRIT based on FBA-PAST outperforms the Adaptive ESPRIT based on PAST in the term of robustness, accuracy and has a lower mathematical complexity.

7. Februar 2014 Olaa Khatib

DECLARATION OF ORIGINALITY

I hereby certify that this thesis was created autonomously without using other than the stated references. All parts which are cited directly or indirectly are marked as such. This thesis has not been used in the same or similar forms in parts or total in other examinations.

Ilmenau, February 7, 2014 Olaa Khatib